Analysis of category co-occurrence in Wikipedia networks

Analysis of Category Co-occurrence in Wikipedia Networks

Thidawan Klaysri

Department of Computer Science and Information Systems

Birkbeck, University of London

A thesis submitted for the degree of

Doctor of Philosophy

September 2019

I would like to dedicate this thesis to the ‘Cognitionis Amor’

I inherited from my father.

Declaration

I, Thidawan Klaysri declare that this thesis is the result of my own work, except where explicitly stated otherwise in the text.

Acknowledgements

I would like to express my appreciation to my supervisors, without whose guidance and persistent help this thesis would not have been completed. I deeply respect Prof. Trevor's intelligence and fabulous research ideas. Following that, since working with Dr. David Weston, I have gained so much breadth of research skills, and I very much appreciate his wealth of support, encouragement and help during my study. He has been an inspiring supervisor, and has been very proactive in responding to my many questions.

I would also like to thank Prof. Mark Levene, Prof. Giovanna Di Marzo Serugendo, Dr. Oded Lachish and Prof. Panagiotis Papapetrou as my co supervisors who gave much advise and support. I appreciate Prof. Pasquale De Meo and Dr. Ida Pu, whose advice and feedback during the viva helped improve this thesis.

Without the support from my beloved husband, Mr. Stephen Roberts and my close friend, Chayanis A. W., I would never have been able to continue my research in the UK. In addition, I would like to thank research fellows I met at the London Knowledge Lab where we had a great time doing our research.

Abstract

Wikipedia has seen a huge expansion of content since its inception. Pages within this online encyclopedia are organised by assigning them to one or more categories, where Wikipedia maintains a manually constructed taxonomy graph that encodes the semantic relationship between these categories. An alternative, called the category co-occurrence graph, can be produced automatically by linking together categories that have pages in common. Properties of the latter graph and its relationship to the former is the concern of this thesis.

The analytic framework, called t-component, is introduced to formalise the graphs and discover category clusters connecting relevant categories together. The m-core, a cohesive subgroup concept as a clustering model, is used to construct a subgraph depending on the number of shared pages between the categories exceeding a given threshold t. The significant of the clustering result of the m-core is validated using a permutation test. This is compared to the k-core, another clustering model.

The Wikipedia category co-occurrence graphs are scale-free with a few category hubs and the majority of clusters are size 2. All observed properties for the distribution of the largest clusters of the category graphs obey power-laws with decay exponent averages around 1. As the threshold t of the number of shared pages is increased, eventually a critical threshold is reached when the largest cluster shrinks significantly in size. This phenomena is only exhibited for the m-core but not the k-core. Lastly, the clustering in the category graph is shown to be consistent with the distance between categories in the taxonomy graph.

List of Notations

Notations	Descriptions
n	total number of vertices
V	the set of vertices in a graph; $\{v_1, v_2, \ldots, v_n\}$
e_{uv}	an edge, a relation from vertex u to vertex v; $\{u, v\}$
m	total number of edges; size of the graph
E	the set of edges in an undirected graph; $\{e_1, e_2, \ldots, e_m\}$
$G=(V,E)$	an undirected graph with the vertex set V and the edge set E; or G^{VE}
a_{uv}	an arc, a relation with orientation from vertex u to vertex v; (u, v)
A	a set of arcs in a directed graph; $\{a_1, a_2, \ldots, a_m\}$
$G=(V,A)$	a directed graph with the vertex set V and the arcs set A
$G=(U,V,E)$	a bipartite graph whose partition has the parts of vertices U and V with edges E denoting every edge, e_{uv} where a vertex u from U is adjacent to another vertex v from V; represented as G^{UV}
d_u	degree of vertex u of a graph
$d(G)$	an average degree of a graph
N_v	a set of neighbours of vertex v
$[A]$	an adjacency matrix
$[A]_{uv}$	adjacent value (can be 1 or 0) of a vertex pair from u to v of the adjacency matrix A
$p(x)$	a power-law, a function of relationship between variable pair where variable x varies as a power of the other

Notations	Descriptions
$F \subseteq G$	a subgraph F whose the vertex and edge sets are subset of graph $G=(V,E)$
clique	a maximal complete subgraph comprising at least three vertices, where every vertices pair is adjacent to each other
k-clique	a standard clique of *k* vertices with a minimum *k* at 3
$\delta(G)$	a minimum degree of a vertex of graph G
$\Delta(G)$	a maximum degree of a vertex of graph G
k-core	a subgraph in which each vertex has at least *k* adjacencies to the other vertices
w	a weight of an edge e_{uv} denoting a strength of relationship between the vertices u and v
e_w	a weighted edge, an ordered pair of vertices with weight w
$\omega(G)$	a minimum weight of e_w of graph G
$\Omega(G)$	a maximum weight of e_w of graph G
m-core	a maximal subgraph in which a pair of vertices is adjacent with a minimum *m* edges
P	a set of Wikipedia pages; $P = \{p_1, \ldots, p_y\}$
C	a set of Wikipedia categories; $C = \{c_1, c_2, \ldots, c_z\}$
isolated page	a page vertex that belongs to at most one category
isolated category	a category vertex that is not sharing any pages with any other category
feeble category edge	a category edge e_w with weight 1, sharing a page in common
G^{PC}	a page-category graph as a bipartite graph representing the network of connectivity between the pages and categories
G^{EW}	a category co-occurrence graph as an edge-weighted graph representing the network of categories, known as co-occurrence graph

Notations	Descriptions
r	a desired subsets number of a set of ordered Wikipedia categories $\{c_1, c_2, \ldots, c_n\}$; n = number of categories
a	a set of ordered categories $\{c_1, c_2, \ldots, c_a\}$ for the first unordered category c_i of a category edge
b	a set of ordered categories $\{c_1, c_2, \ldots, c_b\}$ for the last unordered category c_j of a category edge
R_i	a range of category edge [a, b] where the first unordered category of the edge is in the category set a and the second category is in the set b
nR	a number of the possible category edge ranges
$\mathscr{R}$	a set of the category edge ranges, which each range comprises category sets a and b; $\{R_1, R_2, \ldots, R_{nR}\}$
G_R^{EW}	a category co-occurrence range graph where the cut edges overlapping from different subgraphs of the G^{EW} are organised
t	weight threshold value, number of shared pages for each category pair
$G_{R,t}^{EW}$	t-filtered category graph obtained from the category co-occurrence range graph G^{EW} by the removal of every edge e_w with weight w less than t
category cluster	a connected component where any pair of categories shares at least a page in common
$C(G_{R,t}^{EW})$	the category cluster of the graph G^{EW}, corresponding of the graph G_t^{EW}
$CC(G_{R,t}^{EW})$	a combined category cluster where the category clusters of all different subgraphs within the graph G_t^{EW} are merged
$t1$	weight threshold t before the largest cluster separation
$t2$	weight threshold t+1 at the largest cluster separation
C1	the largest category cluster
C2	the second-largest category cluster
C3	the third-largest category clusters

Table of contents

List of figures

List of tables

Chapter 1

Introduction

The world has become more connected, trackable and networked due to the proliferation of digital and mobile communication. Consequently, the dramatic growth of information, both in content and online social interactions, over cyber networks has been significant in the big data era. [1, 2] It is enormous not only in the content scale, but also in the complexity of its structure [3, 4]. In digital data science communities such as social and behavioral science, bio-informatics and physics, mining this big data has become a great challenge in discovering the relationships between entities [5–7]. Analysis of the big data in terms of complex network science is to understand networks and their formation [1]. To comprehend a complex network, we rather explore its anatomy [8] where latent knowledge can be found. For example, to understand the structure of large complex networks of the web by examining the network's topology, interesting phenomena can be revealed [7–9].

Wikipedia, a popular internet-based encyclopedia, is a noncommercial web social media platform. It is one of the most visited global web sites[1] but has been ranked lower than other social networking sites such as Youtube and Facebook. Wikipedia has had a dramatic growth in the volume of information it contains, and also in the very rich social interactions within

[1] ranked by http://www.alexa.com/topsites (reviewed in May 2018)

its complex networks. The resources in Wikipedia are considered as open big data with the three V's [10, 11]. The first V is the large '*Volume*' of the social interactions constructed from people's contributions and considerable content in articles and categories. The second V is their collaborations among a '*Variety*' of wiki-sources such as multilingual, interlinked and manually categorised data sources (i.e. articles and categories) [12, 13]. The final V is '*Velocity*' of the real time user-generated content [11] (e.g. timestamps of pages [14, 15]). Wikipedia has been particularly attractive to researchers because it is freely available, and has richness of phenomena and resources for investigating, testing and leveraging in various applications. It also has the capability to embrace improvement in analytics over different fields such as social science and computer science [11].

1.1 Background on Wikipedia

The rich open web-resources of Wikipedia such as multilingual content, links structure, manually categorised data sources, concepts and name entities, have for more than a decade helped to foster its attention by the research communities [16]. Wikipedia is perceived as a large public knowledge base [17], where its content is suitable for knowledge base construction [18, 19] such as [12, 20–25] constructing corpora. The prominent utilisation of the Wikipedia category (topic) graphs were to solve entities ranking [4], text clustering [26] and classification [27, 28]. The graph was also used to assess the semantic relatedness of word pairings [29–32], to build a thesauri and ontology [33]. The Wikipedia graph analyses such as [34–37], were used to enhance efficiency of algorithms, improve capabilities of applications and enrich accuracy of text analysis results.

For education purposes, recently [38] has built a knowledge graph from Wikipedia and Google Scholar to help students find their research experts, and [39] contributed a personalised text recommendation framework helping learners to gain more knowledge

from Wikipedia. The awareness of 'bias on the web' (in web use and content) that might cloud web users judgment and behaviour [40] has been raised, such as political, cultural and gender bias in the internet society [41, 42]. The population of the content contributed in Wikipedia may be represented differently from women and men [42], for example [41–44] studied the gender biases in Wikipedia. A brief overview of the encyclopedia, the main components, in particular articles, categories and category system of Wikipedia are presented as follows.

Wikipedia provides multilingual encyclopedias in about 300 languages with wide coverage for various branches of knowledge. The English Wikipedia was the first edition, founded in January 2001. It has 5,661,345 content articles, 45,105,652 pages in total, and averages 600 new articles per day. It has the highest number of usage views per hour, at least about five times higher than the other editions such as German, French, Russian, Italian and Portuguese (see *the link to statistic information*[1]). Wikipedia has a variety of articles, and most articles comprise paragraphs of content, images, tables and references to those sources. [45]

An **article** in Wikipedia is created as a page of comprehensive summary of knowledge about existing topics under the large umbrella covering more than 10 *major subjects* (last reviewed in July 2019) such as 'Culture and the Arts', 'Geography and Places', and 'People and Self'. Content of the article is intended to be encyclopedic information well-written in a formal tone, underlining their three core content policies: 'neutral point of view', 'verifiability' and not containing 'original research' [46, 47]. This content is collaboratively created, edited and modified through a 'wiki', written in mark-up language, as an openly editable model by more than 20 million registered collaborators in Wikipedia, called '*Wikipedians*' [45]. The articles in Wikipedia are as accurate in covering scientific topics as the encyclopedia Britannica [20]. However, "*Wikipedia will always be a work in*

[1] https://stats.wikimedia.org/EN/Sitemap.htm (reviewed in June 2018)

progress, not a finished product", Broughton noted [48]. It has been focusing on improving the quality of content in the articles rather than the quantity, since the articles already cover the most important topics. Also, readers expect good quality content from high reputation contributors [49].

A **category** in Wikipedia is a 'category page' (to avoid confusion, in this thesis notes 'category' is short for 'category page') that links a number of articles under a few common subjects or related topics. There are two main types of categories. The ***content categories*** are part of the encyclopedia, and are provided for the readers to find related articles by topics, which can be either topic categories (named as article on topic) or set categories (named after a class) or a combination of both. The top level of the categories can be found at '*Category:Contents*'. The ***administrative categories*** are for managing non-article pages by the current state of the articles, which are categorised into four types. The first type, 'maintenance category' indicates the modification of the category pages and evokes templates for maintenance of project such as 'cleanup' and 'fact'. The second type, 'stub category' contains articles which are too short to be part of the encyclopedia. The third type, 'wikiproject category' is a group of contributors that aim to improve Wikipedia. Finally, 'assessment category' is used to assess the quality of articles on a particular topic underlying the quality scale, using the 'Wikipedia:Version 1.0 Editorial Team' as the assessment system. In addition, there are other types of categories such as 'file pages' to manage images, video and audio and 'talk or discussion pages' to improve articles; They are supposed to be placed in the administrative categories where appropriate and generally would not appear in the encyclopedia's category navigator. However, there are cases where they appear in the article pages, and should be assigned as 'hidden categories' to hide the maintenance activities from readers. Due to the high volume of categories, Wikipedia has maintained the same persistent rate of article growth, and they are required to be categorised. [50, 51]

The **category system** of the encyclopedia in Wikipedia is where the large amount of the articles are annotated into a number of possible related categories or at least one most appropriate category. Each category can also be placed into multiple categories where a subcategory can be a member of multiple super-categories as the categorisation's guidelines provided in [50, 52]. The lists of relevant subjects are rationalised collaboratively by Wikipedians who have unlimited eligibility to create and modify the annotation to any category. Thus, the category structure has become chaotic [53–55] because the relations between subcategories and super-categories are very much overlapping [12, 48]; A partial categories connectivity is shown at *Wikipedia category network diagram*[1]. Wikipedia provides several tools for readers such as *PetScan*[2] to search articles through subcategories and super-categories and *CategoryTree*[3] to browse the categories.

The categorisation in Wikipedia is a self-organising system regarding the arranged knowledge of the diverse topics [55, 56]. Its system of the indexing articles and categories is characterised as a large collaborative thesaurus, where list of related subjects arecategorised together; It combines the thesaurus and the *collaborative tagging*, e.g. social bookmarking, where tags (category labels) are annotated for the articles. The category associations in the collaborative thesaurus are more connected flexibly than a *taxonomy* such as library classifications; e.g. Dewey Decimal System. [12, 18, 29, 31, 57–63]

1.2 Research Background

Wikipedia is regarded as one of the largest knowledge base sources and is favored when studying by the research community, including this thesis, as interests are in its one of the major resources, the categories. The categorisation mechanism of Wikipedia annotates

[1] https://en.wikipedia.org/wiki/Wikipedia:Categorization#/media/File:Category-diagram.png
[2] https://en.wikipedia.org/wiki/Wikipedia:PetScan
[3] https://en.wikipedia.org/wiki/Special:CategoryTree (These 3 links were last reviewed in July 2019).

an article into a number of most relevant categories and a subcategory into specific super-categories where it would logically belong [48, 50]. Indeed, "*you can't use logic on human behaviour*", quoted, Jeff Lindsay. Although, Wikipedia has launched the policies and guidelines [64] to help users for those contributions, still "*human behavior flows from three main sources: desire, emotion and knowledge*", Plato quoted. An interesting point is that the growth of categories is higher than the articles. For example, the number of pages and categories increased by around 40% and 50% from 2010 to 2012 [65], and 12%, and 25% between 2012 and 2014 [66]. A perspective of Bergman [18] is that "*... the biggest problem of Wikipedia has been its category structure. Categories were not part of the original design*".

The ground truth network of the rich association among the pages and categories that are generated from the Wikipedia category system [12, 31, 32], in this thesis, is represented as a graph, called '***page-category graph***'. When connected categories that are annotated from at least one page, it constructs a new '***category co-occurrence graph***'. In NLP (Natural Language Processing) research, the topics or concepts extracted from this category graph were used for semantical knowledge base construction [67–71] and semantic relatedness measurement [29–32].

This thesis is motivated by investigating the structure of the co-occurrence graph. The graph analysis is challenging due to the scale and complexity of the graph. To illustrate, the English page-category graph edition 2015 contains almost 20 million pages and more than a million categories; There are complicated associations among the pages and categories, about a hundred million links. This graph is indeed too large to be held in main memory. It demands a capable graph analysis methodology to examine the connectivity of the pages and categories and identify *category clusters*, where all possible semantically relevant categories are connected.

In modern network science, traditional random graph theory was challenged by the intensive claims for scale-free topology on real world networks [13, 15, 72–77]. They have indicated that the networks were mechanically growing by attaching new vertices to existing ones nonrandomly selected. This is supported by the well-known *preferential attachment* model of Barabási [78], explained in Chapter 2, Section 2.2. A suggestion about graph modelling in [8] is that "*power-laws*[1] *are not just another way of characterising a system's behavior. They are the patent signature of self-organizing in complex systems*". For a better understanding, in one of the largest knowledge based networks like Wikipedia, a *power-law* can be used to characterise the network. Furthermore, the evidence of scale-free topology existing with a few hubs and networks not growing randomly, relates to the fact that the power-laws have been acknowledged in vast network studies, e.g. in Wikipedia's pages [13, 15, 72–77, 79] ; others [80–95]). However, the hubs are more difficult to identify in social networks than other types of networks [96]. More importantly, the considerations of characterising a network demands more than fitting a power-law to the data, but a method to identify the hubs [8] or at least a suggested region where those hubs would be identified easily. This motivates network scientists to resolve the puzzles, as does the analysis in this thesis to identify the potential hubs in the Wikipedia category graph. To approach that, alternative graph analysis methods such as *cohesive subgroups*, graph clustering models, e.g. *k*-core and *m*-core; more details provided in Chapter 2, Section 2.5 are surveyed and discussed in this thesis.

This thesis investigates the structure of the Wikipedia category co-occurrence graphs where the content and any form of sematic analysis are not concerns. The original data sets are four networks of English from 2010 to 2012 and 2015, three German networks from 2010 to 2012 and four other languages from 2012: French, Russian, Italian and Portuguese. They are downloaded from the *data sets' link*[2]. Each of the data set is the 'CategoryLinks'

[1] A power-law is a function of relation between a pair of observed variables; see Chapter 2, Section 2.2
[2] http://dumps.wikimedia.org (last reviewed in July 2019)

table containing links from article (pages) to category (pages) and subcategory to super-category membership relations.

1.3 Thesis Contributions

The experimental works presented in this thesis contributes to the graphs analysis in Wikipedia. The contributions demonstrate the methodology and reveal the insights of the analysis.

First, this thesis provides a methodological contribution which focuses on the analysis of co-occurrence graphs in Wikipedia to cluster all the categories on a large scale. The *t*-component framework operates on the graphs with two main phases as follows.

1. The ***graph manipulation*** phase in the framework is capable of handling large graphs. It can examine the complex relationship among the pages and categories, and derive the category co-occurrence graph from the page-category graph with three main functionalities:

   (a) The *graph partitioning function* performs on dividing an initial large page-category graph into subgraphs.

   (b) The *graph constructing function* transforms each page-category subgraph into its corresponding co-occurrence subgraph.

   (c) The *graph organising function* guarantees the whole graph will contain unique category edges.

2. The ***graph clustering*** phase in the framework enables clustering for the category graph on a large scale with three main functionalities:

(a) The *graph filtering function* performs on nested filtering by employing the *m*-core model to ensure a minimum number of shared pages between the categories.

(b) The *category clusters identifying function* searches for category clusters.

(c) The *clusters combining function* merges all category clusters in different subgraphs into a single cluster.

Finally, the experiments contribute to the novel findings on the graph properties, graph clustering and insights of the category hubs separation. The contributions are briefly listed as follows:

- The evolution of the structural properties for the examined English Wikipedia graphs from 2010 to 2012 and 2015 are presented and discussed.
- The regularity patterns, i.e. power-laws with the presence of a few hubs observed on the size of the category clusters in English and other five languages examined and the insights of the graph analysis, are presented and discussed.
- A category hub separation phenomenon, where the largest cluster is divided into a few smaller clusters appears in each of the category graphs; This affirms that a few category hubs connect most of the categories. An insight of what caused the fragmentation of the hubs is revealed.
- The finding of the category hub separation is validated on a number of random permutations of the page-category graph whether the categories are connected randomly or not.
- The category hub separation phenomena detected by the *m*-core clustering model is compared with the result for the *k*-core.

- The category hubs result is validated on the taxonomy graph if the category structure of the two graphs is consistent when measuring distances within and between the category clusters.

1.4 Thesis Outline

The background on Wikipedia and its category system, research background and the contributions of this thesis are provided in this chapter. The thesis content covers graph analysis, a literature survey of articles and categories in Wikipedia, a research methodology to analyse the graph structure on a large scale. Also, the experimental work, which contains the main strands of the investigation into the category structure graphs, the results are presented, validated and discussed. The rest of the chapters are as follows.

Chapter 2 presents a background on graph analysis. The content covers the essential graph terminology and graph modelling employed for the graph analysis in this thesis. The fundamental concepts in Social Network Analysis, the cohesive subgroups and in particular the theoretical concepts of the cores such as the k-core and m-core are provided and their applications are also discussed as a justification of the clustering methods.

Chapter 3 is a literature survey on the analyses of Wikipedia. The first survey is the analyses of the Wikipedia pages, where the content of articles and the structure of the page-links were studied. The second is the survey on the Wikipedia category graphs, where the semantic and taxonomy category graphs are reviewed. The final survey covers closely related work on the analyses of Wikipedia category graphs. The various applications of category graph-based analyses, analyses of co-occurrence graphs and graph analytic methodology are discussed.

Chapter 4 introduces a methodology to analyse the co-occurrence graphs on a large scale. The terminology of Wikipedia graphs used in the experimental graph analysis is defined. It is followed by an introduction to the t-component framework which includes the m-core nested filtering approach to cluster the co-occurrence graphs. The two main phases in graph manipulating and graph clustering from the framework are explained and demonstrated. The graph manipulating phase partitions a large graph into sub graphs, transforming each sub graph into a category graph and managing overlapping category pairs among subgraphs. The graph clustering phase is where the sub graphs are filtered, and the category clusters are identified within the sub graphs. Finally the discovered clusters are combined.

Chapter 5 presents the co-occurrence graph clustering results using the k-core and m-core, and reveals the insights of the analysis. The results on the graph properties and the category graph clustering, where the focus of the observation is on the size of the largest cluster, are presented and discussed. The comparison of the clustering results on the k-core and m-core is presented next. Finally, the insights of the category hubs separation are revealed.

Chapter 6 presents how the clustering results, presented in Chapter 5 can be validated. First, the clustering result focusing on the size of the largest cluster is tested on the random permutations of the page-category graphs. The final test is to validate the category hubs result on the taxonomy graph.

Chapter 7 summarises this thesis which covers the t-component framework, the novel findings from the clustering, the insights of the analyses and the clustering validation. In addition, a possible direction for the future research following this thesis is also suggested.

Chapter 2

Background on Graph Analysis

The background on graph analysis used for the analysis of category co-occurrence in this thesis is provided in this chapter. The first two sections cover the essential graph terminology and graph modelling. The next section is the necessary theoretical concepts of Social Network Analysis (SNA). The concept of the cohesive subgroups, in particular the cores such as k-core and m-core and their applications are presented and discussed in the final section.

2.1 Graph Terminology

This section presents essential graph terminology for different types of graphs representing the Wikipedia networks which are used in the analysis of category co-occurrence in Chapter 4 and taxonomic networks in Chapter 6. The fundamental approach to find the connected components used in the analysis methodology in Chapter 4, is also provided.

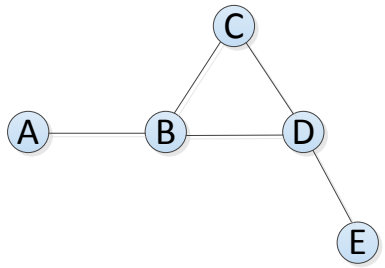


Figure 2.1 An undirected graph

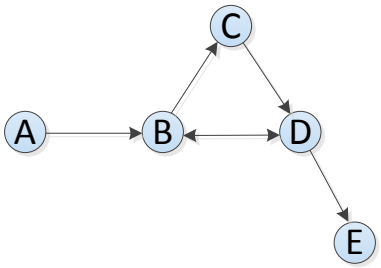


Figure 2.2 A directed graph

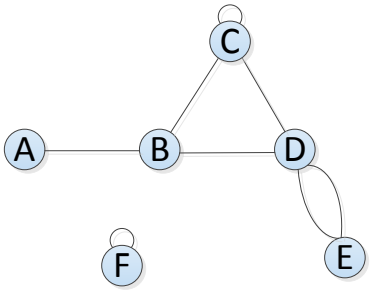


Figure 2.3 A multiple graph

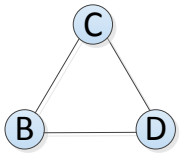


Figure 2.4 A subgraph

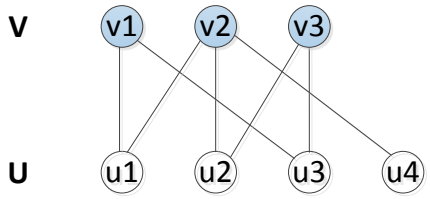


Figure 2.5 A bipartite graph

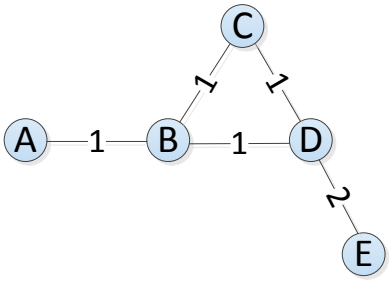


Figure 2.6 A weighted graph

Undirected Graph

An undirected graph G=(V,E) comprises a set of vertices V and a set of edges E, of unordered pairs of the vertices. An edge, e_{uv} connects two vertices u, v, which denotes as unordered pair {u,v}. This graph is presented in Figure 2.1, where there are five vertices, V = {A,B,C,D,E} and five edges, E = {{A,B},{B,C},{B,D},{C,D},{D,E}}.

Directed Graph

A directed graph G=(V,A) containing a set of vertices V and a set of arcs A where an arc, a_{uv} is an ordered pair of vertices adjacent from vertex u to vertex v. Figure 2.2 shows this graph of five vertices, V = {A,B,C,D,E} and five arcs, A={(A,B),(B,C),(B,D),(C,D),(D,E)}.

Loop

An arc or edge is a loop if a vertex is connected to itself.

Multiple Edge

An edge is a multiple if a vertex is joined to the same vertex more than once.

Simple Graph

A simple graph is an undirected graph which contains no loops and no multiple edges. as shown in Figure 2.1.

Multiple Graph

A graph is a multiple graph if it contains either 'multiple edges' or 'multiple arcs', and allows loops. Figure 2.3 is representative of the graph that has the multiple edge {D,E} and two loops {C} and {F}.

Isolate

A vertex is an isolate if it has no relation to any vertices in the graph. An example of the isolate vertex F is represented in Figure 2.3.

Subgraph

If a graph F=(V,E) is a subgraph of another graph G=(V,E), $F \subseteq G$, the vertex and edge sets of F are subsets of G. A subgraph can be obtained by removing vertices and edges from the entire graph. For example, Figure 2.4 is representative of the subgraph obtained from the simple graph as shown in Figure 2.1, where the incidents, {A,B} from A to B, and {D,E} from D to E were absent.

Bipartite Graph

A bipartite graph G=(U,V,E) is composed of two sets of vertices U and V and a set of edges E. Every edge, e_{uv} is an adjacency between u and v where u belongs to U and v belongs to V. The bipartite graph is presented in Figure 2.5.

Weighted Graph

A weighted graph comprises the set of vertices and the set of edges where each edge has a weight w denoting a strength of relationship between the vertices u and v. The multiple graph as displayed in Figure 2.3 can be converted into a weighted graph in Figure 2.6 where each multiple edge is summed as a weight, the loops and isolate were removed.

Connected Component

A connected component is a subgraph in which any pair of vertices are connected to each other by paths. In this thesis, this component is considered as a 'cluster'. Most large undirected graphs in the web, in particular in information and communication networks, have a significant 'largest component' (giant component) where most elements are connected,

and have 'small components' with very small sizes (less than 10 vertices) for the rest of the population. If there is a vertex from each of the large connected components then both components are by definition connected. The size of the largest component is generally greater than 50% to 90%, and it is rare to find a large network that has more than one large component [9, 97].

A simple algorithm to search for connected components, where vertices are connected together in a graph is Breadth First Search (BFS), which will be used within the t-component framework proposed in Chapter 4. The 'queue', *first-in-first-out* data structure is used for the graph traversal from one to all vertices.

Algorithm 1: BFS for finding connected components

Input: G=(V,E)

Output: C, a set of connected components

```
1  visitedV = VtoVisit = ∅
2  repeat
3      connectedV = ∅
4      Dequeue(V) → u_o
5      if (u_o ∉ visitedV) then
6          u_o → Enqueue(VtoVisit)
7      while (VtoVisit ≠ ∅) do
8          Dequeue(VtoVisit) → u
9          u → visitedV
10         u → connectedV
11         for (each neighbour v of u) do
12             if (v ∉ visitedV) then
13                 v → Enqueue(VtoVisit)
14     if (connectedV ∉ ∅) then
15         connectedV → C
16 until V = ∅;
17 return C
```

Algorithm 1 is representative of the search. A vertex u_o is 'dequeued' (line 4) from the `V` set of vertices to search for a connected component by looking for its neighbours (lines 11-13). A condition of the search is that all vertices must have been visited, and each vertex will be visited only once (by marking as 'visitedV'). Each of the unvisited neighbours will be appended to 'VtoVisit', a waiting queue for a next traversal vertex being a member of the 'connectedV', a connected component. To obtain all connected components, the procedures will be repeated until all vertices are visited.

2.2 Graph Modelling

A crucial graph model, 'power-law', appears commonly in large real networks such as Facebook [77], Twitter [72, 76] and Wikipedia [13, 15, 73–75], and used to model big data [72]. The power-law consisting of two laws of network growth and preferential attachment are explained in this section.

Power-law Distribution in the Web Graphs

A ***power-law*** distribution is functioned from the relationship between two observed graph properties such as sizes of subgraph and frequency of multiple edges. When the property's quantity is varying, it gives rise to a proportional relative change in another. The power-law function is written:

$$\mathtt{p(x)} = \mathtt{a}\mathtt{x}^{-\alpha} \tag{2.1}$$

where `p(x)` is a function resulted by varying variable `x`, `a` is a constant and α is a power-law exponent.

$$\log\ p(x) = \log\ a - \alpha \log\ x \tag{2.2}$$

Then in Equation 2.1 we can say that $p(x)$ scales as x to the power α. Let us take Equation 2.1 in a logarithm as presented in Equation 2.2; $\log\ p(x)$ depends linearly on $\log\ x$ with the line's gradient as the α.

A crucial characteristic signature of a power-law is that the plots show long-tailed or heavy-tailed in L curve shape where the distribution decays slowly. This is because there are a higher frequency of the multiple edges in the graph (much larger than the mean) than Gaussian (normal distribution in bell curve shape) or exponential distributions [5, 8, 96]. We can simply identify the power-law by plotting the graph properties' distribution on a log-log scale. If the log-log plots appear as a straight line, the line can be checked for the correct fit to the power-law model by using the coefficient of determination [87, 88, 98, 99]. A good explanation of how to estimate the slope and the goodness of fit for the power-law [100] is referred to readers, and the techniques of power-law fitting are discussed and demonstrated in [72, 77, 87].

A network's degree distribution following a power-law is called a *scale-free* network [13, 96, 101]. The original scale-free phenomenon is that adding new vertices and edges or arcs generates over time a rapid growth in the network [101]. In term of the 'scale-free', for either of the graph properties observed from 100 to 1000 or between 1000 and 10000, their power-law distributions exhibit the same [96, 99].

The power-law can describe a phenomenon where the small frequency of observations are exceedingly ordinary, but the large ones rarely appear such as the network properties distribution in the internet [87, 88, 102]. To illustrate, only a few popular visited web sites in the world wide web such as Google, Yahoo and Amazon, get most of the users's attention.

The power-law distribution appears in various empirical data sets. A crucial finding, for instance, the Pareto distribution of people's income (later called 80/20 rule as the law of factor sparsity) is interpreted that most money, around 80% is earned by a few wealthy people, roughly 20 % of the population. Another power-law found is Zipf (cumulative) distribution of a quantity, for example the frequency of words in documents is plotted against their ranking, known as 'rank/frequency' plot [8, 85, 87, 99]. The power-law in literature always referring to the laws of Vilfredo Pareto and George Kingsley Zipf may cause a confusion. The difference between the two distributions is the plots of Zipf had the variable `x` on the horizontal axis and `p(x)` on the vertical axis, while it did the other way around for Pareto's [87, 88]. In addition, most real scientific and man-made networks such as the topology in the internet [103], web pages in the world wide web [104] and large social networks obey power-law distributions [8, 80, 105].

The power-law distribution comprises two laws. The first law is the phenomenon of a network's '***growth***' in exponential scale at a time where each new vertex has been added into the network. The exponential growth in most networks generated from web applications in the internet obeys power-law distributions with varying exponents which tend to fall between 2 and 3 [96, 99]. The second law is a '***preferential attachment***' mechanism [78] Barabási Albert model; A real network can grow naturally when new vertices have been added to the graph and attached preferentially to high degree vertices. This explains the network's growth, when the older vertices have a high probability to be attached from new input vertices, consequently, connected vertices have expanded as a larger component [7–9, 86–91].

Most large complex networks contain hubs, where a *hub* is a highly significant amount of vertices that are well connected. The existence of the hubs indicates a nonrandom network and exhibits a relationship of degree exponent between the most and less popular vertices [7–9]. The presence of a few hubs connecting a large number of vertices together is a crucial

property expected in the scale-free network and not in random graphs [8, 96]. Identifying hubs in large networks are useful in diverse domains such as ensuring the internet's robustness against failures [84] and spreading news through social networks [106].

2.3 Graph Clustering

There are many clustering algorithms to find communities or related instances in networks. This section provides a brief discussion on a few clustering algorithms, in order to give a justification and a more concrete idea of what other methods of graph clusterings there are.

Clustering is an approach to group data instances together where they share a few properties in common or have a similar pattern of their associations [107–109]. For data mining perspective, the clustering is generally used to identify regularities or patterns within the (attribute) data using a wide range of techniques from classical statistics to data mining, regarded as an unsupervised learning of discovering and summarising the clusters without labeling [5, 37, 107–112]. The clustering principle is that each cluster contains similar members, having a closer relationship (with high similarity) than the outer clusters (with low similarity or high distance between clusters) [37, 107, 108, 113]. The clustering techniques can be categorised into (1) hierarchical clustering and (2) partitioning [114] (or flat) clustering.

Graph mining is an approach to discover unknown knowledge or to detect regularity in a network using a graph to represent relationships [5, 100, 115]. Most of web graph mining concerns graphs of linked entities (e.g. hyperlinks), and these web graphs can be examined to determine interesting dense communities [116], see [117] for a survey of graph mining applications and community detection, see also [116] to mine and manage graph data.

Graph clustering is a crucial task of the graph mining to commonly analyse a graph by partitioning the set of vertices into clusters (or communities in social networks) using various measurements such as vertex similarity (e.g. cosine similarity in text clustering) and vertex connectivity (e.g. weights of edges or path length between each vertex pair) [5, 118–120]. Following the clustering principle, a cluster of similar vertices (subgraph) must have a higher weight of edges, and lower weighted edges among different clusters [37, 107, 108, 113, 120]. A recommended survey of methods for graph clustering to readers is [119]. There are various well-known graph clustering approaches such as partitioning relocation, hierarchical and spectral clusterings, reviewed very briefly as follows.

(I) Partitioning relocation clustering

Partitioning relocation clustering algorithms are widely used in numerous applications, and have been improved for decades, a classic example is k-means [121–125], a recommended survey is '50 years beyond k-means' [126] and a good review of k-medoids [127]. A crucial difference between these two algorithms is how their centre points are chosen for clusters when partitioning. A centre cluster of the k-means is computed from an average of data points, while k-medoids uses an exact centre from those data points [127, 128]. When comparing their performance, [129] showed that performing k-medoids can reduce the time in computation of adjusting the random index of the medoids. Considering their average performance, k-medoids performed better for large data sets than k-means [128].

(II) Hierarchical clustering

Hierarchical clustering algorithms were deployed for various types of networks. For example, [130] grouped sensors in wireless networks, and [131] reconstructed a hierarchical cluster using textual and link analysis. In social networks, [132] presented personalised recommendations in social tagging systems, and [133] improved blogosphere annotating by reconstructing

a topical hierarchy among tags. Furthermore, the clustering algorithms have been widely used for document data sets. To illustrate, using the hierarchical clustering in Wikipedia, [134] constructed a tree structure of concepts extracted from Wikipedia's documents, and [135] clustered tweets using Wikipedia concepts. In fact, [136, 137] demonstrated that agglomerative algorithms produced lower hierarchical solutions than the partitional methods. In information networks such as Wikipedia, the knowledge bases generally were represented in taxonomy manner where the concepts and topics are organised in the tree hierarchies such as [4, 12, 18, 26, 58, 138–144]. However, "*it depends on the application and the input data whether it makes sense to compute a hierarchy of clusterings or a flat clustering*", Schaeffer stated [119]. Indeed, this thesis does not focus on constructing or presenting category clusters in the form of a taxonomic category graph.

(III) Spectral graph clustering

Spectral graph clustering algorithms borrow concepts from spectral graph theory using the eigenvectors of a similarity Laplacian matrix of a (weighted) graph, see the graph Laplacians in " tutorial on spectral clustering" [145]. Basically, the matrix is mapped from the original similarity, e.g. (raw) weighted graph in adjacency matrix to the eigenvalues space [113, 114, 146, 147]. Spectral methods are simple to implement and efficient [145, 148] for the clustering in different (even large) data formats [145, 149], and can produce nonlinear separating hypersurfaces between clusters [114]. However, their computations are generally demanding, and also have the issue of choosing the number of k clusters [119, 148]. Furthermore, there are a few concerns of the spectral methods: constructing a good similarity graph that must ensure a sparse graph, and choosing appropriate parameters for the neighborhood graphs [145].

2.4 Social Network Analysis

This thesis is researching for an analytic approach to cluster the categories and analyse their structure in the Wikipedia co-occurrence network. In SNA, clustering is a way of discovering and summarising groups of actors by their relations' structure. The analysis in this thesis focuses on vertices of pages and especially category, and is not concerned with any of the other attributes which can describe the pages and categories. In graph mining, clustering is commonly used to identify regularities or patterns within the attribute data using possible methods such as statistic and/or machine learning [107, 112]; Also, an attribute describing the vertices can also be used to measure their similarity in graph mining. Whilst, structural relationship of vertices are generally concerned in SNA [117, 150]. Besides, in social and behavioral science, network clustering is to partition the actors for a clearer understanding of the actors' behaviour with little or no prior knowledge about the network [108]. There has been a vast amount of research that uses social network analysis to find communities in various social networks such as [151–159].

This section provides a brief introduction to SNA to cover the main ideas of social network's representation. An overview of the essential terminology related to social network analysis, and the representation of social network data will be employed for the proposed analysis methodology in Chapter 4. They are explained as follows.

A **social network** is, in general, a network of 'actors' which are connected by their 'relations'. The actors or network's members can be people, organizations and other observed entities and the relations or 'ties' are a representation of the actors' relationships such as 'friendship', 'kinship', 'partnership' and so forth. This network can be represented as a graph. Generally, cooperative networking in social science focuses on the relationships between people rather than the context in which they interacted. For example, in citation networks,

how authors collaborate (e.g. who is connected to whom) is of more concern than the content of what they are publishing in the web [160–162]. Considering the web, which contains networks of information such as page-links indicating relationships of the pages [163], how they are grouped into similar subjects, and which ones are most prominent [164]. The associations of the web pages can be considered as a social network of the authors contributing the pages [99], even though, the subjects are categorised by the content in the pages. There are also communities of web users, whose contributors generate documents through web social networking. Indeed Wikipedia which is the focus of this thesis, is another good example of a social web community of Wikipedians, creating and modifying the wiki-pages in the encyclopedia.

SNA is a paradigm to study relationships of actors in a network. It is used to acquire regularity or connectivity patterns of a network and reveal implicit behaviour related to the connectivity structure [161, 162, 164–166]. Mathematical metrics from *graph theory*[1] are used to model or measure relations in the network that can describe relationship patterns or regularity of vertices connectivity. There are two types of analysis approaches:'ego-centric' focuses on an individual or specific actor that leads to the others, and 'socio-centric' takes a network population into the analysis [165]. With the SNA approach, we can answer questions such as, who the members of an observed community are, and how much they interact to each other [167]. For instance, the analysis [168] identified behaviour patterns of Wikipedians' editing in English and Japanese Wikipedia. The interactions among the pages and categories in the Wikipedia categorisation system can be seen as an example of a social network that infers to their editors. Analysing the links of categories can show which categories are connected, by how and how much they relate to each other by looking at their structure.

[1] using discrete mathematics, which combines many disciplines such as geometry, probability, set, combinatorics, geometry and theoretical computer science to solve problems in applied sciences such as physics, social science, and furthermore in economic studies [161, 165]

For an analysis on a large scale, the examples of structural properties that would be considered are component sizes, largest component size, density, path lengths and degree distributions [9]. Generally, a small network tends to have a high density, whereas, large networks are more sparse. One of the most essential metrics is 'centrality' telling us which actors have more connections than the others. The analysis can also be concentrated on the strength of weak ties [169] where they join one subnetwork to another.

Social network data contains at least a structural property which can be used to measure the relationship of actors [160]. There are two types of data to be determined: *attribute data* describes characteristics or descriptions for individual actors such as age, gender, location, salary and so on, whilst, *relational data* presents the connections between a set of actors in pairs no matter what the descriptions for each individual actor would be [162, 170]. *Which one of the two data types is more suitable to construct the social network?* In some ways, the actors can be connected by a specific relation such as 'contacting to', 'employed by', 'clicked by' and so forth. A relation represents how the actors are connected in common not the characteristics or descriptions of their individuality. In this context, the relational data is rather an appropriate type of data to present the social network [170].

Social network data can be represented in a *relational data table* as showed in Figure 2.7. A social network may contain more than one set of actors as a bipartite graph of a page-category links network shown in Figure 2.8, in which SNA is termed as a 'two-mode network' or 'affiliation network' [160, 161]. A network with a single set of actors is called 'one-mode network' like the category-links network, only remaining category pairs are represented in a relational data table (Figure 2.9) that is transformed from the page-category links graph (Figure 2.7).

Page List	Category List
p1	c1
p1	c2
p1	c3
p2	c3
p2	c4

Figure 2.7 A relational table of the page-category links network

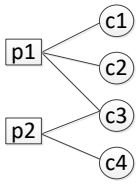


Figure 2.8 A graph representative of the page-category links network

Category List1	Category List2
c1	c2
c1	c3
c2	c3
c3	c4

Figure 2.9 A relational table of the category-links network

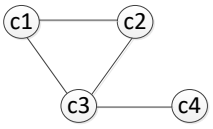


Figure 2.10 A sociogram, representative of the category-links network

	c1	c2	c3	c4
c1	-	1	1	0
c2	1	-	1	0
c3	1	1	-	1
c4	0	0	1	-

Figure 2.11 An adjacency matrix of the category-links network

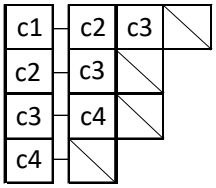


Figure 2.12 An adjacency list of the category-links network

$$[A]_{uv} = \begin{cases} 1 & \text{if} \quad e_{uv} \in E \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

Let graph G=(V,E) be a simple graph of category links, where V is a set of categories and E is a set of the links where $[A]_{uv}$ (see Equation 2.3) are adjacent values of the category pairs in the adjacency matrix [A]. The matrix denotes category links from one to another category (u to v).

An **adjacency matrix**, [A] is a $n \times n$ matrix, which represents the graph in matrix form. Figure 2.11, for instance, shows a symmetric matrix with 4×4 columns and rows for four categories and 16 elements for the adjacencies between the category ties. The element $[A]_{uv}$ of the matrix [A] is 1 if there is a direct connection between categories u and v, and 0 otherwise (see Equation 2.3). It can be seen that c1 is connected to c2 and c3, c2 connected to c3, c3 connected to c4, but c1 and c2 are not connected to c4. In practice, representing a network in a form of a complete data matrix has a memory cost that scales as the square of the number of vertices. Therefore, storing the matrix as a list (as shown in Figure 2.12) would be a more practical way to represent this for large networks. This list of edges is called an **adjacency list** where the neighbours of each vertex, $N(v); v \in V$ are listed in some order such as, $N(c1) = \{c2, c3\}$ and $N(c3) = \{c1, c2, c3\}$.

The 'sociogram' is a network diagram, invented by Jacob L. Monreno in early 1934 to visualise a network using symbols and lines such as a point (vertex) for an actor, a line (relation) for an edge/arch. For instance, the sociogram in Figure 2.10 is a representation for the adjacency matrix and list in Figure 2.11 and Figure 2.12. However, it has a limitation when depicting much larger and more complex networks.

2.5 Cohesive Subgroups

A cohesive subgroup is a subgraph of connected vertices where their connectivity can be quantified by vertices degree and frequency of edges among vertices or multiple edges count [160]. The cohesive subgroups are regularly considered as an undirected graph, but 'role' and 'position' are more interested in digraph [171]. This section provides the concepts of cohesive subgroups such as cliques, and in particular the cores, which will be applied in Chapter 4 for the analysis methodology.

Cliques

A **clique** is a maximal complete subgraph of a graph comprising at least three vertices, *triad* or clique of size three where the 'complete' subgraph has every distinct vertices adjacent to each other [9, 160, 166]. The clique is concerned with the 'complete mutuality' of the vertices [160]. Cliques are usually considered as dense subgraphs [172]. The models can represent the intense relationship of people's groups where they have a few unique common bonds, such as religion, ethnicity and especially coauthors in a citation network. Various applications use them to model the networks such as bioinformatics and network engineering [160, 173]. A clique in which the largest shortest path length between any two vertices does not exceed k is called a k-clique, a 'standard cliques' of size k, in which the k value indicates the density of vertex members in the cluster [174, 175]. In graph theory the standard clique (called 'tightly knits' in SNA) is described as a 'cluster concept' [176], and considered as a 'standard cluster model' [173]. The k-clique was used to identify dense communities on dynamic social networks, relations of messages in ENRON and co-authorship articles in DBLP in [159].

On the theoretical ground that a clique allows overlapping vertices between groups, it seems to be a wise choice to detect a social community as used in [172, 174, 177]. The clique model was applied to extract disambiguation name entities in NLP application on AIDA, a dataset aggregated from Stanford NER, YAGO2 and English Wikipedia [178]. It was also used to categorise the Wikipedia's name entities [179]. This clustering model was not only used for structure-based analysis in Wikipedia network [180], but was also used to leverage the content-based applications on Wikipedia sources [181–185].

In practice there are a few concerns about clique models that have a less practical clustering method in real world networks, in particular for the man-made networks due to the restrictive cohesion requirement. This is because the clique size depends on the size of

the complete graph [162], and vertices of the cliques overlap each other, which also belong to the other cliques [166]. Bolikowski [180] stated that "*in theory, each connected component should be a clique and cover one topic. However, incoherent edits and obvious mistakes result in topic coalescence, yielding a non-trivial topology*". In terms of the clique model restriction, Evans [186] explained that "*it has been argued that considering only complete subgraphs is too 'stingy'* ". To avoid bias when partitioning vertices, he demonstrated a constructive 'clique graph' representing the overlapping communities from a weighted graph. Fortunato and Claudio [187] gave the model's justification that "*triangles are the simplest cliques, and are frequent in real networks; Larger cliques are rare, so they are not good models of communities. Besides, finding cliques is computationally very demanding, ... The definition of clique is very strict*". The clique has the restrictive cohesion requirement of forming a maximal complete subgraph [160, 173, 188]. In fact, the empirical research found that the social networks in the web have a low density, and it was not often required to obtain the large size of the cliques. Therefore, it needs to expand the maximal shortest path length. For the clique restriction, Balasundaram [173] made a crucial point that the standard clique was soon found to be overly restrictive and impractical. Primarily this is because real-life clusters do not meet the 'ideal' notion of cliques. Therefore, to relax the clique requirements of the maximal shortest path, the core concepts such as the k-core and the m-core are alternative considerations in clustering models.

2.5.1 Cores

A **core** is a maximal subgraph of either a directed or undirected graph which is not necessarily a complete graph. It is rationalised as a cluster model based on the quantifying of vertices degree or weighted edges [160]. The concept of the *core* is to construct sets of subgraphs by increasing a threshold t. Consequently, a few vertices would be disconnected, and

the entire graph will be divided into smaller subgraphs. These subgraphs of the current core can be filtered to obtain a set of denser (with higher threshold) subgraphs by increasing the threshold value until it meets a threshold convergence or a maximum degree or weight. The cores can be reconstructed continuously to obtain nested cores, where each core has a set of maximum subgraphs by setting a parameter to quantify the cohesion of vertices within the subgraphs. This iterative constructing of the subgraphs is called '*nested filtering graphs*'.

To illustrate, the parameter k (in k-core) is for vertices degree and m (in m-core) is for weighted edges. The next core can be obtained by increasing the parameter such as $k+1$ or $m+1$, and filtered from the previous k and m cores. Each core can be obtained by removing any vertices that have the degree less than the k or weight less than the m. Removing the vertices causes a few ties and the other connected vertices to be absent. This is how the subgraphs within the core are filtered. At this stage, the clustering is performed to obtain clusters within the core. The BFS as presented in Algorithm 1 can be utilised to identify the clusters or subgraphs. The cores are simple to construct and have been seen widely in social networks for searching communities and clustering vertices in various scientific networks.

The k-core and m-core methods have similar concept to construct the cores by filtering the subgraphs. The distinction between these two core approach is that the k parameter in the k-core by Seidman [189] is parallel to the m parameter of the m-core, Scott [170] termed; The m stands for 'multiplicity' of ties to measure intensity between the vertices. Whilst, the k parameter is the vertices degree quantifier in the k-core. These two core approaches will be discussed and explained individually.

In SNA, the core concept is classified to be a 'relational strategy' to define a 'boundary specification' of a relations population when studying social networks such as

the popular k-core method [162]. The core concept was introduced to give more flexibility for connections among vertices in the maximal subgraph. Identifying subgraphs can depict an area of 'global clusters' of the whole network and a few other interesting features, such as 'local clusters', which could be investigated further [160]. The concepts and the clustering algorithms of the k-core and m-core, are explained, and a review is made on how they are used in graph analysis.

2.5.2 *k*-core

A ***k*-core** is a maximal subgraph where each vertex is adjacent to other vertices with a minimum degree of k. The 'degree' of a vertex `v`, `d(v)` is the total number of vertices adjacent to `v`. The number k is the parameter that restricts the 'strength' of each subgraph's members. The range of k is from a minimum vertex degree (δ) (starting from `2`) of every vertex in the core, to the maximal degree of the vertices (Δ) (maximum is `n-1`) [173].

Algorithm 2: k-core nested graphs filtering

Input: Original graph `G=(V,E)`, minimal degree δ and maximal degree Δ

Output: Subgraph `F=(V,E)`$=\{\mathtt{F} \subseteq \mathtt{G} \mid \delta \leq k \leq \Delta\}$

```
1 The first core's all subgraphs F = the original graph G
2 for each k-core from k = δ + 1 to Δ do
3     A next core's subgraph F_{k+1} = previous core's subgraph F_k
4     for each vertex v in V of subgraph F_{k+1} do
5         V of subgraph F_{k+1} - v where d(v) < k
6     return union subgraphs F^{VE}_{k+1} of current core
```

The concept of the k-core, one of the most popular methods to find a dense subgraph, is based on the quantifying of 'vertices degree'. The core concept with the k parameter is to extract the dense and its weaker connected vertices around it. The k-core model identifies maximal subgraphs by repeatedly removing vertices having degree (number of neighbours)

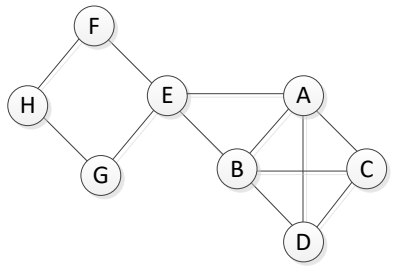


Figure 2.13 A simple graph

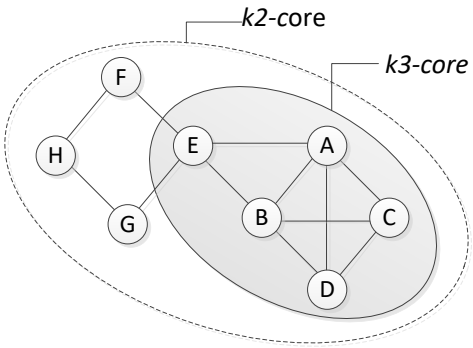


Figure 2.14 *k2*-core: {A,B,C,D,E,F,G,H} and *k3*-core: {A,B,C,D,E}

less than k. The 'k-core nested graph filtering' is demonstrated in Algorithm 2 influenced by 'k-core decomposition' [190], which runs in linear time with respect to the number of edges. It constructs a graph for every value of k, where the k value is not greater than the maximum degree and not less than the minimum degree, i.e. $\delta \leq k \leq \Delta$. The algorithm requires for input: an original graph G=(V,E), minimum degree δ and maximum degree Δ of vertices in the graph. For a vertex v, the total degree can be obtained by $\sum_{u=1}^{n} e_{uv}$. The sizes of the cores, however, decrease, and decrease rapidly for large sparse networks. Algorithm 2 (lines 4-5) shows that the union subgraphs F_k^{VE} of a core with minimum degree at k can be obtained by removing any vertices that have the degree less than the k. Removing the vertices for each current k-core causes a few ties and the other connected vertices to be absent. Increasing the k value, new subgraphs, the $(k+1)$-core are obtained from the

previous k-core, Algorithm 2 (lines 2-5) shows this. Note that the k in the algorithm begins with 2 (k=1 is the original graph). BFS in Algorithm 1 can be added into Algorithm 2 after line 5 to identify the natural clusters in the form of connected components. This whole process is called 'k-core clustering'. For details of computing and sorting vertices' degrees for the k-core, see [191].

The k-cores where the k value is equal or greater than 2 and equal or less than 3, ($2 \leq k \leq 3$) are illustrated in Figure 2.14 which are obtained from the graph in Figure 2.13. It shows the two k-cores with two different numbers of the k in the range of 2 to 3. For each core any vertex v that has the degree from any vertex u (N(v)) less than the k will be removed, so that any edges and vertices linking to the vertex v will be also absent from the core such as {F,G,H} in the $k3$-core (k=3). For the first core, the subgraph is filtered up with k=2 as the lowest degree δ, and the result is {A,B,C,D,E,F,G,H}. It has the same result as the $k1$-core which is the original graph.

2.5.3 Applications of the k-cores

The k-core is considered as one of the practical clustering models for graph analysis because it is more flexible and much simpler to construct than the cliques [173]. The method is commonly used in large scale network analysis in various applications reviewed as follows.

In SNA, the k-core method is used to select a population of vertices when studying social networks [162]. For instance, it was employed to expand the snowball sampling of the social networks to define and locate the cores and boundaries of the networks [192], and also used to select landmark vertices for a network's compact routing [193].

The following studies in bio informatics data used the k-core clustering. A predictive method based on the k-core [194] was used to construct subgraphs within protein-protein

interaction networks. The $k3$-core gave significant predictions of unknown proteins, and was evaluated with a random graph having the same size as the empirical graph. In mass protein spectral data as a weighted graph [195], was clustered on a large scale by filtering the weight threshold using 'P-CAMS' as a parallel algorithm to run on multi-core machines. Genes of the gene networks from PubMed data [196] were clustered using the traditional k-core with DFS to identify connected components, and the genes cluster results were ranked using TFIDF: The combined methods obtained the clusters with more control in the cluster sizes, and had negligible overlapped clusters.

In co-authorship networks, [151] used the k-core clustering to identify and rank the dense co-authorship collaboration in the DBLP, computer science bibliographic publications network. The initial network is in an undirected bipartite graph of associations between the papers and authors, and was converted to a weighted graph where each weighted edge presented a relationship between a coauthor by using the number of co-published papers. The dense subnetworks of the collaborated author communities in the undirected co-authorship graph in [152], was approached by using the k-core clustering in the Pajek package [197]. An optimization framework was proposed in [198], which was able to obtain the top ẽmphk-core members (most tightly connected together), and revealed their relevant relations.

In other social networks, such as a mobile network [191], the k-core was used to find a dense subgraph in a distributed manner. The k-core model can derive a privacy-preserving protocol, and search the optimal k-core to ensure the protocol security. An analysis in a blogs network [199] employed the k-core as a methodology to identify a community in the blogs. The social hypertext structure such as triangles, where people shared emotion in the blogs was analysed: The hypertext model demonstrated that the social behaviour could be comprehended automatically without doing a behavioural survey.

'*K*-core decomposition' introduced by Batagelj and Zaversnik [200] is a centralized algorithm developed from the traditional *k*-core of Seidman [189]. The new *k*-core is to find the *k*-core of the graph for every possible vertex degree *k*. It runs efficiently in linear time, `O(m)` when constructing the hierarchy of the graph. There has been a large amount of recent graph applications commonly using this algorithm to analyse and visualise complex networks where most of them were treated as unweighted graphs. For example, [201] the structure of the internet was modeled, [202] protein complexes in the protein-protein interactions data were detected, and [203] the central part of the earthquake and the structural properties of the earthquake network were reported. Besides, for document classification [204] used the *k*-core to improve the performance of the classifier. In addition, it has also been used to approach NP-hard problems on empirical networks such as [205–207].

A distributed *k*-core decomposition on large dynamic graphs [193] was used to select a set of vertices and exploit the graph structure to yield a compact routing internet protocol. An algorithm to update the number of cores for every vertices in a dynamic graph was presented in [207]. The *k*-core algorithm with game theoretic approach [208] was employed to model vertex engagement dynamics in many large social graphs by measuring at the vertex level and examining properties of the graphs.

There has been attempts to extend the capability of the *k*-core decomposition for certain situations. The *k*-core composition algorithm [209], was extended to the weighted graphs (e.g. protein-interaction networks) where a cluster of protein is obtained by iteratively deleting the vertices that have a degree less than a setting value. An incremental updating process added to the algorithm for streaming graph data to run faster when deleting and inserting vertices, and the need not to traverse the whole graph was developed in [205].

A new 'distributed *k*-core decomposition' [210] provided two computational models: the one host-one vertex evaluated with the Standford large network and the one host-many

vertices employed with the Amazon EC2 which could not run on a single machine due to limitation of the memory. The algorithm [190] that was improved from [210] was proposed by running a large graph in the internal memory on a single machine. However, when a graph is very large (i.e. about up to one billion edges and fifty million vertices), it had an in-memory issue. A solution for that was the EMcore, an external-memory k-core decomposition algorithm [206]. The model required the maximum k value of the graph.

Even though the entire semantic category graph of the Wikipedia English 2015 version examined in the anlaysis of this thesis can be held as relational graph-based data, the growth of the network's size was a concern. If the network cannot be operated in-memory, the core clustering algorithm running in external-memory such as the EMcore would possibly be considered further. The partitioning technique to handle the large graphs in the t-component framework was demonstrated by operating the smaller pieces of graphs divided from the whole large graph, which is available for future growth of the network.

2.5.4 m-core

In scientific networks, the k-core approach has been used for various graph applications. By the weighted graph definition that is a representation of a multiple graph, where the multiple edges can be presented as the strength of the edges, these weights can indicate clusters' intensity. An example in logistics network applications, when establishing a new path into the logistic routes, the weights can be analysed to manage the cost. It is possible to approach the problem by using k-cores. However, in many cases we consider '*how often vertex u is connecting to vertex v?*', showing intensity more than '*how many other vertices that u is connected to?*'. Instead of only using the degree of vertices to measure the density, the strength of relations are more concern; Taking the co-authorship network as an example, the graph was treated as a weighted graph where the number of co-published papers was

assigned as the weight representing the relationship between each coauthor. An alternative core method, m-core, which is based on the multiplicity (multiple edge count) of vertex pairs can approach those.

A ***m*-core** is a maximal subgraph where each vertex is adjacent to other vertices with a minimum m edges (that each pair of vertices share). The core is constructed from a weighted graph that is generally represented for a multiple graph where the number of multiple edges are summed and determined from the relations' intensity. The parameter m indicates the threshold of the weight value (i.e. connectivity strength) to restrict the strength of the connected vertices in the subgraphs. The m value can possibly begin from a minimum weight of the edges ω to the maximum weight Ω or the opposite.

Algorithm 3: m-core nested graph filtering

Input: Original graph G=(V,E), ω and Ω

Output: Subgraph F=(V,E)= $\{F \subseteq G \mid \omega \leq m \leq \Omega\}$

1 The first core's all subgraphs F = the original graph G
2 **for** each m-core **from** $m = \omega + 1$ **to** Ω **do**
3 A next core's subgraph F_{m+1} = previous core's subgraph F_m
4 **for** each vertex v in V of subgraph F_{m+1} **do**
5 V of subgraph F_{m+1} - v where $d(v) < m$
6 **return** union subgraphs F_{m+1} of current core

The 'm-core *nested graph filtering*' is demonstrated in Algorithm 3. It constructs an m-core graph for each value of m, where the 'm' value is inbetween the minimum and the maximum weights of the edges ($\omega \leq m \leq \Omega$). The input requires the highest weight of all edges in the weighted graph is Ω and the lowest weight ω starting from 1 (the original graph), 2 (having weight threshold at least 2), 3, 4, ..., Ω-1. Note that, in Chapter 4, Section 4.4.1 for the m-core nested clustering, the ω is set to 2 and the edges having weight 1, are ignored. These are weak ties or feeble edges, see Definition 5.

The *m*-core is considered as nested structural clustering, a similar approach to the *k*-core where the cohesion within the cores is increased as the number *m* decreases. The cores sizes, however, decrease, and decrease rapidly for large sparse networks.

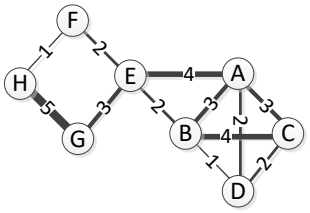


Figure 2.15 A multiple undirected graph

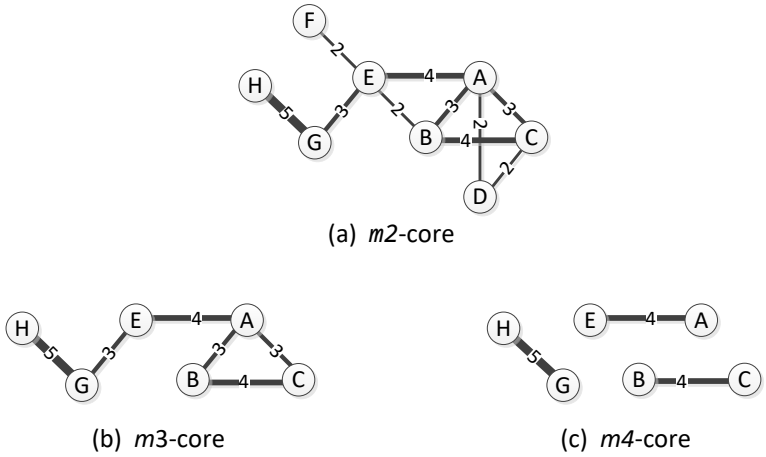


Figure 2.16 *m*-cores

Algorithm 3 (*lines 4-5*) shows that the union subgraphs `F=(V,E)`, notated as $\mathtt{F}_m$ where $\mathtt{F} \subseteq \mathtt{G}$ of a core can be obtained by removing any edges that have the weight less than the *m* value. Removing these edges causes a few vertices and the other connected vertices to be absent. For each current core, the *(m+1)*-core is filtered up, and the subgraphs are obtained from the previous *m*-core, Algorithm 3 (*lines 2-5*) shows this. The BFS in Algorithm 1 can be added into Algorithm 3, after line 5 to identify clusters in the form of connected components. This whole process of the '*m*-core clustering' will be deployed in the *t*-component framework in Chapter 4.

The original graph as showed in Figure 2.15 is an example of a core that has only one connected component, considered as a *m1*-core, where the minimum weight threshold *m* is equal to 1 and maximum weight at 5. Figure 2.16 (a) shows that any edges with weight less than 2 were removed such as the edges {F,H} and {B,D}. However, the cluster's size or the size of connected component remains the same as the original graph in Figure 2.15. Figure 2.16 (b) shows that any incident ties with weight less than 3 were removed such as the edges {E,F}, {B,E}, {A,D} and {C,D}. The size of the cluster in this core is smaller than the previous *m2*-core where the vertices D and F are absent as they had incidents from the multiple edges having weight less than 3. The result of *m4*-core is presented in Figure 2.16 (c) where the single cluster from the *m3*-core has split into three clusters with every vertices connected to the others with weight ≥ 4. A current *m*-core can be obtained from the previous *m*-core, for example, the *m4*-core in Figure 2.16 (c) is a filtered graph by removing some edges from the *m3*-core as shown in Figure 2.16 (b), and so on.

The *m*-core is based on '*q*-nearness' in '*q*-analysis', a mathematical methodology, which is a geometric approach for structural analysis, and is based on algebraic topology [211, 212], introduced by Ron Atkin in 1974 [213], 'mathematical structure in human affairs'. The analysis aims to analyse structural properties of networks based on relationships among sets of observations for categorical analysis which is considered as a qualitative method [212]. In cluster analysis, the *q*-analysis focuses on clustering objects, not clustering in attributes, and derives from vertices sets known as 'simplices' and edges as 'shared faces' of the simplices. A multiple dimensional matrix can represent the relationships such as the incident matrix of the multiple sets of vertices and their interactions. This matrix presents '*q*-nearness' of the edges which quantify the multiplicity as the weight [211, 212, 214]. The analysis is used to analyse the dimensional view of structural relationships within data and solve complex systems such as medical images in the representation of a data matrix or data vector. It can be used in various applications, for instance, in decision making systems

such as [215] modeling the decision making. In social network applications such as a citation network, [216] used q-analysis to analyse friendships of the social scientists. The analysis was utilised to study the interactions among the users such as [217].

2.5.5 Applications of the m-cores

A real world network might be comprised of more than one set of vertices such as an affiliation network, where its properties are not easily interpreted. Hence, the networks [153, 218–222] were derived as a weighted graph representing a one-mode network, concerned with only a single vertex set from the bipartite graph. The m-core model was used to study different types of affiliation networks, which were represented in bipartite graphs such as economic networks in [218, 219, 223, 224]. The filtering technique in the web-link structure mining [219] where only the strong hyperlinks remained, is similar to the m-core based on the multiplicity of edges (amount of ties connecting same pair of vertices).

A nested filtering technique, termed 'm-slice', the m means 'Matreyshka' as the nested Russian doll or 'slice' in short [225], which has a similar concept as the m-core, was used widely for hyper media application design and development in Relationship Management Methodology (RMM) [225, 226] such as [225–228]. The associations within hyper media in the web is more complex than other software designs, because it involves the navigation of relations and objects, user interface and information processing. Therefore, the m-slice method extended the RMM for hyper media application design [227] is more suitable than an ad-hoc design. This is to reduce the cost of providing guidelines for project managers and developers and to select attributes from different entities [229]. The m-slice technique attempts to represent the relationship between entities (breaking many-many into one-many relationship) in an E-R diagram (Entity Relationship diagram) into a slice, which is reusable by enabling an iterative combination of top-down and bottom-up sub-processes as stated in

[230]. In web-based information systems such as [226], a framework of RMM that can gather information from heterogeneous sources either relational or Object Oriented Databases was proposed.

The *m*-core has been applied to many diverse fields of work such as software development, education, medical and biology. It is called '*m*-slice' in few applications. The knowledge network, an affiliation network of software development teams where developers exchanging information and communication though Apache web server was investigated in [220]. Research [231] used the *m*-core where the *m* value was set to 5 to clearly highlight the software patent conflict. The *m*-core was approached as a collaborative filtering network structure of online interactions in e-learning [221]. The method was used to filter and personalize the learning environment such as rearranging groups of learners and topics. In the medical field, [222], for example, the *m*-core model was used to identify clinical archetypes by finding relationships in the structure of terms which was represented in a bipartite graph constructed from the Unified Medical Language System meta-thesaurus. The cohesive method was also used to extract the largest connected component, the giant cluster of the random gene promote network in [232], and the growth of the network was studied via the preferential attachment. A bipartite graph of Book-crossing network was examined by employing the *m*-core in [153] to seek a potential strong group of users in the web community. The Chinese co-authorship networks were analysed by using *k*-core and *m*-core, focusing on: oncology subject in [233] and collaboration on the cardiology and cardiovasology fields in [234]. The collaborated groups result obtained via the *m*-core was superior to *k*-core on the co-authorship network because the weight of the co-authors plays an important role for the cooperations in the networks.

The *m*-core model was also used in economics and some other applications. A study of economics, see '*the structure of networks-the transformation of UK business*' [223] in pages 48-65 used the clustering model (provided in the Pajek package [197]) to analyse the British

interlocking directorates network (boards of directors of the 250 largest UK companies). The counts of each company pair indicated the relationship joined by two directors, and the company groups within the *m2*-core were obtained and the largest cluster was focused on revealing the business evolution. [218] studied evolution of the telecommunications in New Jersey and Texas over the period 1996 to 1999 focussing on networks of inventors. Each of the networks was converted from the bipartite network of the relationship among patent assignees and patent inventors into a weighted graph representing the one-mode network of the inventor network by using Pajek and the m value was set as 2 for the model. Research [219], the m-core and the Apriori association rule were applied to identify the most frequent market association patterns in the relationship between enterprises, and their export markets were represented in a bipartite graph.

The cohesive models that have been reviewed so far, found that there are a few considerations for the models justification among the k-cliques, k-cores and m-cores. The network scale is the first concern. The computational costs of constructing the clique and plex models made them hard to use compared to the k-core models, which have contributed more in approaching hard problems, and are far more simple to implement. Whilst, the m-core model is commonly used in weighted networks such as the co-occurrence networks (e.g. co-authorship networks). Determining between the two core methods for the analysis of the Wikipedia category graphs in this thesis, the m-core is chosen for the structural clustering model. This is because the principle of constructing a co-occurrence graph is more in regard to quantifying the number of shared pages for a connected category pair (i.e. category edges' multiplicity), and less concerned in quantifying the number of neighbor categories allocated to each category (i.e. category's degree). The collaborated groups result of the co-authorship networks analysis [233, 234] obtained via the m-core model was better than the k-core because the shared works between the co-authors plays an important role for the cooperations in the networks more than the number of individual

work for each author. Although, we can see the superiority of m-core to the k-core, the co-occurrence graph represented as an information network is different from co-authorship.

For the category graph analysis, the shared pages of each category pair are considered as a similarity or connectivity's strength. However, each category in a category pair can also be joined to other categories by other pages. This co-relationship can be considered as a probability value or a ratio of the number of common pages shared between two categories. In this thesis, the clustering results from the m-core will be compared to k-core.

Chapter 3

Related Work

The brief background on Wikipedia, its main components (i.e. article and category pages) and its category system was presented in the first chapter. In this chapter, the useful background material for the analysis in this thesis is provided in the first two sections. The first section reviews the studies in article pages of Wikipedia. The second section is the survey on Wikipedia categories where the taxonomy and semantic category graphs are applied to enhance various algorithms and leverage the capability of applications. The final section is a comprehensive review of related work to this thesis on the analyses in Wikipedia category graphs.

3.1 Survey on Analyses of Wikipedia Pages

Although, the analysis in this thesis does not focus on article pages, it is worth reviewing where it fits in the broader fields of the Wikipedia mining community. This section provides a brief survey on the text analysis in the wiki-pages and page-link graphs analysis. The scale-free topology which was reported in the page-link graphs are also reviewed.

Text Analyses in the Wikipedia Pages

There has been many studies in Wikipedia's content. For knowledge base construction, [12, 20–25, 235] derived the text corpora from the documents, where a thesaurus defining the semantic relationship among words [57, 142] can be constructed. Regarding where Wikipedia provides multilingual articles, [24] compared the potential translation candidate terms in English and Italian, and [25] improved coverage terms for the Japanese and English editions.

To deal with the large volume of the documents, [20, 33] used the link structure mining technique when constructing the thesauri from Wikipedia's hyperlinks. The technique also showed significantly better accuracy results than the NLP methods such as 'n-gram model', a sequence of n words [236, 237] and the TF-IDF, Term Frequency-Inverse Document Frequency [238]. For text classification, the classifiers [142, 143, 239–241] predicted a words' complexity, which were built based on word frequencies from the Wikipedia text corpus. In addition, to improve the performance of a documents modelling, [26–28] have all been enhanced by deriving knowledge from Wikipedia. This is to overcome the limitations of 'BoW', Bag-of-Words model, a content-based method for text classification by quantifying the appearance of terms or words, where semantic relations are discarded.

Page-link Graphs Analyses

Graph-based analysis uses a graph to represent a network of association among entities without any text consideration such as the analysis of the links structure of the pages [15, 72–74, 76, 77, 242–249]. DF-Miner [249] constructed domain-specific terms from the hyperlinks structure of Wikipedia pages, which performed better than the content-based approach. A technique to analyse the graph is *link analysis* such as the earlysuccessful webpage ranking algorithms, PageRank [250] and HITS [251].

The PageRank and HITS algorithms were applied in many applications such as [247] estimating the content quality of articles by observing the page-link graph structure. Alternatively, [245] used MapReduce distributed parser as a crawler methodology to capture the massive semantic relationship among the page-links; Another approach is SNA (i.e. cliques and cores) as reviewed comprehensively in the previous chapter. For example, [242] used a combination of the k-core and k-clique, called the k-dense to discover the largest connected components in the page-links graph and studied the page-links and time evolution of the number of pages. In this thesis, to reveal the analysis insights of the category-links in the category co-occurrence graph, *what appropriate methodology would be used*?

The Wikipedia page-link graphs can be used to leverage the text analysis, such as to enhance efficiency of algorithms, improve an applications capability and enrich accuracy of text search results [34–37, 249]. For example, the connectivity of the pages were used to rank entities [4], perform text clustering [26] and document classifying [27, 28]. The page-links were also used to assess the semantic relatedness of word or concepts pairings [29–32, 252, 253] such as building thesauri and ontology [33]. To improve text analysis for word sense disambiguation [254] used co-occurrences of page-links in the Wikipedia corpus instead of BoWs or the articles' link distribution.

Scale-free Page-link Graphs

The hyper links among the pages in Wikipedia have been studied where the crucial real world graphs phenomena like the power-laws were revealed. For example, [13] reported that the page-links of the German edition is a scale-free graph where the pages' distribution of in-out degrees follow the power-laws. Also, [242] reported that the distribution of the number of pages and page-links follows a power-law with an increase exponent of `1.30`.

Capocci et al. [73] analysed the page-links for several language editions represented as directed graphs in their growth, topology and degree distributions. Their major finding is that each graph topology exhibits the bow-tie structure indicating a scale-free graph [104, 255, 256] as do other web graphs [104, 255]. In and out degree distributions obey power-laws with a decay exponent falling between 2 and 2.20; The strongly connected components of the pages are discovered; The average size of the largest component for those editions is around 80% of all pages.

The analysis of Zlatić et al. [74] reported the scale-free characteristic with preferential attachment behaviour of edge propagation for the page-link graphs in many language editions. Apart from the Polish and Italian that showed the discrepancies of the graph structure are caused by the contributions of Wikipedians on the calendar pages via wiki-templates. The analysis on those former graphs revealed that their in and out degree cumulative distributions obey the power-laws for exponents within the interval 2 and 3. The average size of the largest connected (pages) component for those graphs was close to [73], roughly 90%. In this thesis, *would the Wikipedia category co-occurrence graph, be a scale-free graph and also for multiple languages*?

3.2 Survey on Wikipedia Category Graphs

The network of the categories generated from the Wikipedia categorisation explained in the first chapter can be represented as a category graph, in short the graph is called 'WCG' (Wikipedia Category Graph). This section has no aim to present a comprehensive survey. It provides a brief overview of applications using the WCG where the rich semantic relationship between pages and categories can be extracted from Wikipedia. A very short background on the taxonomic category graphs and related applications to the taxonomy constructed from the categories are also provided.

3.2.1 Semantic Category Graphs

As noted in the first chapter, the category connectivity in the category co-occurrence graph is determined from the page-category graph. This category graph is deliberated as a *semantic category graph* where semantic relations among categories are concealed [29] and semantic information can be obtained by extracting category association [257].

Many applications in NLP and IR (Information Retrieval) concern the *semantic relatedness*, measuring how many words or concepts using any types of lexical similarity. This is to define coverage association that would appear between two words, such as 'cars-gasoline' and 'night-dark' rather than semantic similarity like a synonym, 'automobile-car'and a hyponym, 'vehicle-car' [31, 32]. The WCGs were utilised to measure semantic relatedness and to improve its assessment's performance by measuring the path length of the category connectivity [29–32]. To deal with the limitation in computing the semantic relatedness of the words using the lexical resources, [30] used category connectivity from the WCG, and [31] adapted the WCG to WordNet. Finding the shortest path length between vertices that [31, 30] did, has inspired me as to how the category clustering results obtained from the category co-occurrence graph (i.e. semantic category graph) could be evaluated on a taxonomy graph in this thesis (in Chapter 6).

3.2.2 Taxonomy Category Graphs

The fact is that Wikipedia contains a good source for new taxonomy topics in both structured data (i.e. category-links) of the category tree and unstructured data (i.e. content) of articles. In the research community of text processing, Wikipedia's topics are more preferable as a taxonomy, and there are many contributions on transforming the category hierarchy into a tree [12, 58]. A categories relationship was used in text clustering [26], entities ranking [4], text classifying [142] and question retrieval [143]. There are many studies using WCG

in IR community to leverage an applications capability and enhance algorithms [18, 143]. For example, to improve the precision of the search results in a web search engine such as [144], taxonomy was used to group the text results.

This thesis investigation concentrates on the category relations connected semantically by Wikipedians and is not concerned with its hierarchy. However, when evaluating the clustering result on the category co-occurrence graph (in Chapter 6), the taxonomic graph [258] will be used as a benchmark category graph. Therefore, the WCG's structure needs to be clarified. The WCG is not a simple taxonomic tree [12, 63]. Ponzetto and Strube [29] stated that "*the Wikipedia categories do not form a taxonomy with a fully-fledged subsumption hierarchy, but only a thematically organised thesaurus. ... the category structure is neither a tree nor a directed acyclic graph...*", explained, Kittur et al. [53]. The WCG comprising multiple parents and loops should rather not be described as a tree and can be represented as a directed graph [58]. Instead, [63] determined it as an undirected tree graph to analyse the differences between its structure and the Universal Decimal Classification. The studies reviewed here give an influential idea of modifying the taxonomy graph [258] to be the form of co-occurrence graph when evaluating the clustering result.

3.3 Related Work on Category Graph Analyses

This section presents comprehensive studies related to this thesis. First, the relevant category graph analyses in Wikipedia [31, 53, 55, 66, 180, 259] are surveyed for what insights the research community have found. Next, [259, 260] the close relevant work on constructing the category co-occurrence graph is surveyed. Finally, [29, 31, 53, 55, 66, 67, 242, 261, 262] the methodology involving the analysis tasks during the co-occurrence graph analysis of this thesis are reviewed and discussed.

3.3.1 Category Graph Analyses in Wikipedia

The analysis results of [55] showed that the average degree of each page has not changed much from 2001 to 2007 and the growth of number of articles, categories, page-links and category-links have a similar trend. Bairi et al. [66] presented the evolution statistics of the English Wikipedia page-category link networks between 2012 and 2014:The growth of categories, articles, category-links, page-category links and admin-categories or administrative categories all increase by around 25%, 12%, 40%, 24% and 10%, respectively. Using the pattern matching rule from [67] to identify admin-categories, there are about 10% of all categories, and the number of admin-categories increased by around 10% from 2012 to 2014 the admin-categories. Interestingly, they found the admin-categories covering approximately 70% articles. The WCG analysis [53] revealed that the most annotated topics in Wikipedia are 'Culture and the arts' and 'People and self' including the popular subjects such as musicians and sport players.

Recalling that the power-law from the previous chapter has two laws of growth and preferential attachment with the existence of hubs indicating a scale-free topology for a graph. The traditional preferential attachment function is understood to be linear [8, 91]. In fact [90] presented that almost fifty online networks follow a nonlinear preferential attachment model, and most of the networks have power-laws exponent much below the expected range of between 2 and 3. The analysis [259] of the category co-occurrence graph revealed that the distribution of the number of category edges per page showed the power-law with exponent 2.96. The graph analysis for NLP application [31] reported that the German WCG has a scale-free topology regarding its degree distribution and follows a power-law. Also, its several graph properties observed are similar to WordNet's such as power-law exponent (i.e. 2.21 for WCG and 3.11 WordNet), average degree and shortest path length. The WCG is claimed to be a suitable source to estimate sematic relatedness between words.

The graph analysis [180] on inter-language links of English Wikipedia reported the distribution of the connected component size of articles and categories follow the power-laws and the graph has scale-free topology. After discovering the cliques of articles, the pages skeleton graph is obtained from filtering the cliques; Their degree distribution also obeys a power-law with exponent 3.75.

Kittur et al. [53] mapping the distribution of topics in Wikipedia found "*that the distribution of categories among pages is not homogeneous. While most categories are distributed approximately equally, the 'People' category is an outlier in having over 2.5 times as many category assignments per page as other categories*". A power-law distribution would probably be a good explanation for the quotation.

The presence of a few hubs containing a large number of vertices, where the vast majority of clusters are small appears in most large complex networks. Barabasi et al. [96] stated that "*no matter how large and complex a network becomes, as long as preferential attachment and growth are present it will maintain its hub-dominated scale-free topology*". Newman [97] analysed scientific collaboration graphs. The relative size of the largest cluster (giant component), containing a large fraction of the related authors was found around to be 80% or 90% of all authors, and has a size far larger than the second-largest cluster's. Bairi et al. [66] revealed some categories were deleted (e.g. empty/renaming categories) *caused the number of categories dropped*; furthermore, there have been *splitting of a category into more categories*.

3.3.2 Category Co-occurrence Graph Analyses in Wikipedia

A brief overview about co-occurrence graph analysis, category co-occurrence graphs and category co-occurrence graph analysis in Wikipedia is provided.

Co-occurrence analysis is a quantitative pairwise study of observed elements to summarise their frequency of occurrence, which is quantified as a co-occurrence data formation [263, 264]. In text analysis, for example, terms co-occurrence were defined when a pair of terms occur in the same document such as the short texts [265] and the Medline database [264]. In social network, the relations among tags from the social bookmarking site [266] and the linkages between science disciplines of the journal co-citation [267] were also analysed as a co-occurrence graph. For author co-occurrence graph, [268] analysed contributors who participated in the same articles of Reuters' news. Also, [97] analysis in scientific collaboration graphs. "*Two scientists are considered connected if they have authored a paper together*"; This defines a graph of co-occurrence scientists.

As this thesis focuses on the associations between the pages and categories, the category co-occurrence graphs constructed in [259, 260] are surveyed for how the connectivity of the categories was interpreted. The category analysis in this thesis is closely related to the research of Holloway et al. [259] in terms of deriving the co-occurrence graph. Two categories can be linked together if they have a page in common. The shared pages frequency of each category pair is assigned as a raw weight to indicate their connectivity strength. The cosine similarity, a metric to measure entities relationship, introduced by Gerard Salton [269] is known as a normalised frequency. It gives better results than the raw frequencies [267, 270] and is used to estimate how close categories are together. The cosine is used to measure the graph structural similarity [271] such as relationship among documents [142, 266, 272–275] and semantic topics coverage [262], and the titles relationship of articles and categories [261]. An alternative of estimating the relationship between the categories

was presented [260]; Szymański used the articles in a page-links graph to normalise the weight of category edges by taking the number of articles linked to each individual category in the WCG into account, and the weight of the category edge was recomputed and placed into the edge.

3.3.3 Co-occurrence Graph Analysis Methodology

This thesis intentions are to discover category clusters in the co-occurrence graph entirely and analyse its structure, but not to perform prediction and classification. The analysis also focuses on vertices of pages and especially the categories, but the other attributes are not a concern. The fact that the scale of the Wikipedia category-links networks analysed in this thesis is a constraint of concern for the analysis methodology choice; *What would an appropriate approach be*?

As stated before, the hubs connecting most of the vertices in a graph are difficult to identify [8, 96]. As explained and discussed in the previous chapter, graph filtering is a simple technique to do so, by deleting edges having weight less than a threshold convergence. The popular k-core model is widely used in analysing various complex networks because it is simple and performs fast such as in various applications [151, 152, 191, 194–196, 199]. The k-core's principle of filtering a graph is that the k parameter is defined for quantify vertex' degree. Whereas, the fundamental of the category co-occurrence concerns weighted category edges where a count of shared pages is assigned for each category pair. Therefore, m-core model, concerning edges frequency such as [153, 218–224, 232], is rather the appropriate choice.

The closely relevant graph analysis in the Wikipedia categories are [53, 259, 260, 262] in which the category structure was examined, and the category graphs were derived from the page-category graph. The category analysis in this thesis is partly related to the research

of Holloway et al. [259]. There are a few crucial differences. Most importantly, research methodology on category clustering was not their prime; Whereas, this thesis introduces an original analytic methodology, the *t*-component framework (presented in Chapter 4) that enables clustering. Another difference, a maintenance category rule-base was constructed to eliminate the non-content categories, and it has much the similar keywords as the work [29, 32, 276]. To be more precise, a few other keywords that indicate assessing quality, grade and importance of articles (e.g. 'ListClass' and 'importance') were also filtrated manually; But, Holloway et al. [259] was not concerned about cleaning the mix type of categories, neither in the process of extracting content categories.

To achieve the thesis goals, there are several tasks during the analysis such as partitioning a large graph, identifying category clusters, and cleaning the mixed types of categories, the research relevant to these tasks are reviewed and discussed as follows.

3.3.4 Page-Category Graph Partitioning

Graph partitioning is used for solving many optimisation problems such as design of very large scale integrated circuits, and managing disks and transportation [277, 278]. For example, a spectral graph partitioning algorithm was used for co-clustering documents and words represented in a bipartite graph [279], and [280] used a spectral algorithm to enhance the classifiers' learning. Alternatively, mapReduce/hadoop (see [281]) was used to operate the massive (web) graphs such as Gbase [282], a graph management and mining system and Pregel [283], a computational graph model, in a distributed computing manner. There are a few challenges when one is partitioning a graph: "*... how to divide a graph into k parts with approximately identical size so that the edge cut*[1] *size ... is minimized*", quoted, Lu et al. [284]. A good graph partitioning algorithm should be able to divide vertices into non-overlapping partitions and minimise the edge cuts [284, 288].

[1] a circumstance that a vertex in a partition is connected to another vertex in a different partition [284–287].

There have been substantial studies to leverage graph partitioning algorithms. For instance, the partitioning model approached the document clustering problem [288] on a bipartite graph allowed a vertex to be partitioned into multiple clusters to relax *hard clustering* (an element can only belong to a partition). The partitioning technique for a graph with billions of nodes which is based on a distributed memory system [284] focused on improving the capability of the graph processing architecturally. Regardless of '*... distributed systems need many machines in a cluster in order to provide reasonable performance*'; [289, 290] contributed parallel graph processing in a single machine.

METIS, a software tools for partitioning large graphs proposed by Karypis and Kumar, provides libraries to partition a large graph into subgraphs, see the materials provided at *web page-Karypis Lab-METIS* [1] [278]. The algorithms are based on multiple graph partitioning by collapsing vertices and edges to reduce the size of the graph, and then uncoarsen it back to the original graph [291]. The multiple k-way partitioning where k indicates the partition at which vertex `v` belongs, is presented [292]. There are two constraints for the graph partitioning functions, to equally size the graph partitions and to minimalise the edge cuts [291, 292]. Many studies on graph partitioning refer to the METIS to enhance efficiency of partitioning algorithms such as [286, 293–297]. K-way partitioning using METIS is fast [292], and it could be as an alternative method for partitioning the page-category graph in this thesis.

The t-component framework has been designed to deal with millions of nodes, and graph partitioning is a first algorithmic concern in this thesis. There are many ways to handle a large graph. At first, a graph sampling method such as the extension of snowball used in [192] was considered for that. Notwithstanding the concern of sampling, a few important category connections might be missing such as a category linking between two hubs.

[1] http://glaros.dtc.umn.edu/gkhome/views/metis (last reviewed in July 2019)

Aside from a few graph analysis tools such as *Pajek* [197] and *Neo4j*, a graph database management system providing various functionalities for large graph analysis are worth considering [298] (last reviewed July 2019).

However, there are many tasks on the basic graph analysis in this thesis such as conducting the category pair's strength and also manipulating the category pairs, handling graph scale, identifying category clusters category hubs. Developing an analytic methodology is preferable to using those graph analysis tools. To make this simple, the paradigm of divide and conquer and the concept from METIS are applied to handle the large graph by dividing it into subgraphs, operating each subgraph, and merging the result. An alternative graph partitioning approach [299] used core composition to partition a graph by filtering the degree vertex, and this would be a wise choice when the co-occurrence graph is too large to process in memory.

3.3.5 Clustering Wikipedia Categories

In the perspective of defining the category co-occurrence graph, Holloway et al. [259] is the most relevant. Nevertheless, VxInsight [300], a graph analysis tool was used to identify the clusters instead of developing a clustering algorithm. Also, [55], Suchecki et al. have not contributed any clustering algorithm. Instead, they used the modularity measurement by calculating the difference in size of actual cluster and random graph to group articles into different category clusters.

Partly similar to this thesis's analysis is [53] Kittur et al. presented an approach to map two different forms of the category relations in WCG : One is its nature form as the *large collaborative thesaurus* that articles are placed into category labels (tagging). Another is a tree structure that is constructed from the Wikipedia's category hierarchy

using BFS traversal that starts from the *category top-level*[1], the link to an example of the top-level of categories (current version, last reviewed in March 2019). A distribution of each article assigned into the category labels is computed and used to classify it into the category tree graph. Also [66] did similarly, but less hierarchical topics. The BFS used in [53, 66] can also be simply used to group related categories together in this thesis.

Although, Suchecki et al. [55] is also interested in categories evolution like [53], their work did not use the category relationship from the Wikipedia category system to confine a tree graph; The article links in the 'PageLinks' graph was used to extract article pages in the page-category graph similar to Szymański [260], to reveal evolution of the content categories. Differently, to perform clustering, Suchecki's identified connected articles from the Pagelinks graph.

Yamada et al. [242] used the k-dense proposed in [301], where the k-core is used to enhance the restriction of the k-clique (complete subgraphs of k) to detect page communities in the page-links graph represented as a simple graph. They analysed the category-links graph using PageRank and HIT to rank the categories in Wikipedia.

In addition, there are other related works on clustering in the WCG: Structural analysis [31], Zesch and Gurevych identified category connected components, and the analysis concentrates on the largest component using DFS traversal to detect category cycles. A document clustering technique [261], Yu et al. generated category subtrees to perform clustering on different features such as category title, relationship of titles of categories and articles and relationship of category title and article text.

[1] https://en.wikipedia.org/wiki/Category:Main_topic_classifications

3.3.6 Cleaning Wikipedia Categories

As mentioned previously, the WCG from the category system is a conceptual network of semantic relations between categories and comprises categories of content and administrative types. A category cleaning process is required when deriving a taxonomy of 'is-a' relationship from the WCG [29, 67, 30]. To construct the taxonomy from the category graph, Ponzetto and Strube [29, 67] removed the non-content categories by searching for particular keywords of the administrative categories in the category names: 'wikipedia', 'wikiprojects', 'lists', 'mediawiki', 'template', 'user', 'portal', 'categories', 'articles', 'pages' [29] and a few more keywords such as 'disambiguation', 'redirect', and 'stub' [67]. Similar keywords were also used in [32, 276]. The approach of Strube [67] was also used to identify the administrative categories in [66].

To present a map of content categories, Pang et al. [262] removed non-content categories by using a few keywords in the 'Wikipedia' namespace to identify administrative categories; While the stub and list categories were identified by a few keywords in the category names such as 'stub' for the stub categories and 'births', 'deaths' and 'History of' for the list categories. Alternatively, a few research [55, 259, 260] used links in the article graph (i.e. page-links network) to identify the content category in the category graph.

The administrative categories are removed from the original category graph to test the category hubs separation phenomenon found in this thesis. The most influential works in cleaning categories are [29, 67, 262] because constructing a rule-base extraction is simple to implement and append and modify the keywords. The keywords and the others that Wikipedia gave can be added into the rule-base such as 'maintenance', 'Hidden-categories', 'Tracking-categories', 'Container-categories', and 'Very-large-categories'. Furthermore, the categories containing a few keywords indicated as 'assessment category' (assessing quality, grade and importance of articles) are also added in.

Chapter 4

Research Methodology

This chapter introduces a methodology to analyse Wikipedia category co-occurrence graphs, where categories are derived from the relations of the pages and categories. The first section provides relevant definitions. The second section introduces the *t*-component framework. This section states the objectives of the proposed framework, and gives a brief overview of the methodology's components. The third section demonstrates how to deal with a large graph. Sections 4 and 5 deal with the actual construction of co-occurrence graphs. The next section discusses the graph clustering phase where the subgraphs are filtered, the category clusters are identified within the subgraphs, and the discovered clusters are combined. The final section is the chapter summary.

4.1 Definitions of Wikipedia Graphs

The Wikipedia graph definitions given in this section are used in the *t*-component framework, which is introduced in the next section as a graph analysis methodology. The definitions of the Wikipedia graphs used in the *t*-component framework are given:

Definition 1 `(Page-category graph)`

Let $P = \{p_1, \ldots, p_y\}$ *be the set of* y *Wikipedia pages and* $C = \{c_1, c_2, \ldots, c_z\}$ *be the set of* z *Wikipedia categories. Each page* $p_i \in P$ *belongs to at least one category* $c_i \in C$.

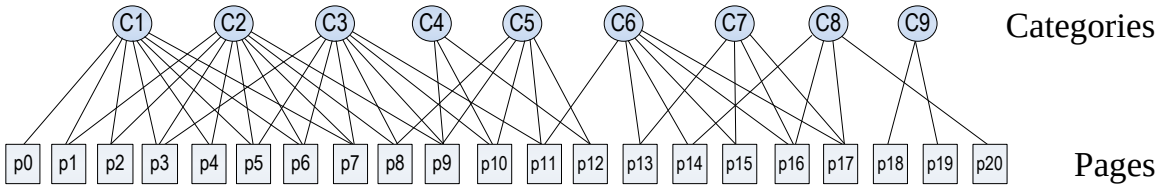


Figure 4.1 A page-category graph

A page-category graph is a bipartite graph G^{PC} *with no multiple edge that represents the graph of connectivity between Wikipedia pages and categories. The set of vertices is* $P \cup C$, *and there is a unique edge* $p \rightarrow c$ *whenever page* $p \in P$ *belongs to category* $c \in C$. *The graph can be seen in Figure 4.1.*

Definition 2 `(Isolated page)`

An isolated page is a page vertex that belongs to at most one category such as the `p0, p18, p19` *and* `p20` *in Figure 4.1.*

Definition 3 `(Isolated category)`

An isolated category is a category vertex that is not sharing any pages with any other category (see `c9` *in Figure 4.1).*

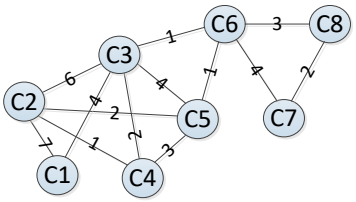


Figure 4.2 A category co-occurrence graph

Definition 4 (Category co-occurrence graph)

A category co-occurrence graph G^{EW} *is an edge-weighted category graph where the set of vertices corresponds to the Wikipedia categories* C*. Each edge* e_w *between two vertices* u *and* v *(corresponding to categories* c_i *and* c_j*, respectively) is assigned with a weight* $w \in \mathbb{N}$ *equal to the number of common pages in both* c_i *and* c_j *(*$e_w = c_i \rightarrow^w c_j$*) or equivalently (see Figure 4.2).*

$$w = |\{p \in P \mid p \in c_i \text{ and } p \in c_j\}| \tag{4.1}$$

Definition 5 (Feeble category edge)

A category edge e_w *with weight* w *equals 1 is called a feeble category. The category graph in Figure 4.2 contains three feeble category edges such as* $c2 \rightarrow c4$*,* $c3 \rightarrow c6$*,* $c5 \rightarrow c6$*.*

Definition 6 (Category co-occurrence range graph)

Given two subranges r_a*,* $r_b \in \mathscr{R}$ *where* $\mathscr{R}$ *denotes a set of possible subrange pairs (and possibly* $a = b$*), the category co-occurrence range graph* $G^{EW}_{r_a,\, r_b}$ *or* G^{EW}_{R} *in short is the new edge-weighted category graph containing precisely those weighted edges* $e_w = c_i \rightarrow^w c_j$ *in* G^{EW} *for which* $c_i \in r_a$ *and* $c_j \in r_b$*.*

Definition 7 *(t-*`filtered category graph`*)*

A t-filtered category graph $G^{EW}_{R,\mathrm{t}}$ *is obtained from an edge-weighted category graph* G^{EW} *by the removal of every edge* e_w *with weight* w *less than* $t \in \mathbb{N}$*, such as* e_w *is in* $G^{EW}_{R,\mathrm{t}}$ *if and only if*

$$w \geq t\ ,\ \forall e_w \in E \tag{4.2}$$

Definition 8 (`Category cluster`)

A category cluster $C(G^{EW}_{R,\mathrm{t}})$ *is a connected component of an edge-weighted category graph* G^{EW}*, and is obtained as a connected component of the corresponding t-filtered category graph* G^{EW}_t *by the removal of all feeble category edges* $(w=1)$ *for a specified threshold* $t \geq 2$ *.*

Definition 9 (`Combined category cluster`)

A combined category cluster $CC(G^{EW}_{R,\mathrm{t}})$ *is that category cluster of all t-filtered category subgraphs* $G^{EW_t} = \{g^{ew}_1, g^{ew}_2, \ldots, g^{ew}_R\}$ *are merged.*

4.2 *t*-component Framework

A research goal of this thesis is to comprehend the structural relationship of categories in the co-occurrence graph. This research also aims to identify all possible category clusters in the form of connected components and investigate how clusters's properties are related to the weight threshold. Then the clustering results will be validated with a taxonomy graph to test if the two graphs are consistent. However, the co-occurrence graph needs to be derived from the Wikipedia page-category graph, which is large in size, and has grown intensively in recent years. To achieve this, a novel methodology to explore the category connectivity which is capable of manipulating and clustering the large graph, is proposed.

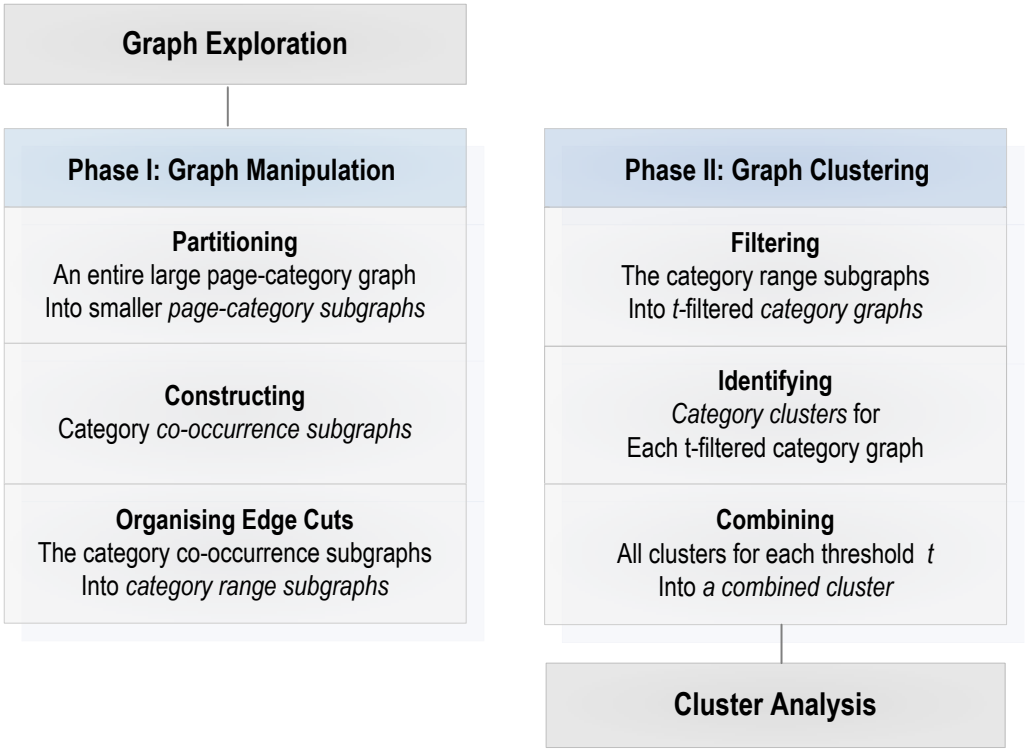


Figure 4.3 *t*-component framework

The *t*-component framework, an analytic methodology is introduced here. It provides functionalities to analyse large bipartite graphs as shown in Figure 4.3. The first process is exploring graphs, to summarise the properties of the large bipartite graph. Before manipulating the graph, one needs to know the necessary properties of the graph such as total number of vertices, edges and isolates of the graph. After performing the clustering phase, the structural properties of the graph and the cluster sizes for different weight thresholds are observed and analysed.

Note that the two terms used in this thesis when handling the graph scale need to be clarified; 'Partitioning graphs' is denoted as a technique to divide a graph into subgraphs to handle edge volumes in the original graph, while 'clustering graphs' stands for identifying category clusters in the form of connected components.

The framework comprises two main phases as follows:

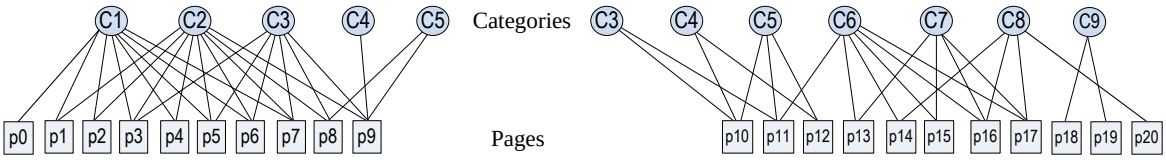
1. The ***graph manipulation*** phase involves transforming a large bipartite graph by
   (a) Partitioning it into smaller subgraphs due to difficulty in operating the entire graph.
   (b) Constructing category co-occurrence subgraphs by transforming each bipartite subgraph.
   (c) Organising edge cuts by assigning the each category edge in the category subgraphs into the co-occurrence range graphs to ensure that there is only unique edge within the whole category graph.
2. The ***graph clustering*** phase employs m-core clustering (see Chapter 2) to explore the graph structure by:
   (a) Filtering graphs: a t-filtered graph is obtained by retaining just a single edge linking each pair of vertices for which the weight of the edge exceeds some specified threshold t.
   (b) Identifying category clusters in the form of connected components where the vertices are naturally clustered together.
   (c) Combining the clusters, the identified clusters are combined into a single cluster.

4.3 Graph Manipulation

The graph manipulation phase consists of three main procedures, partitioning an entire page-category graph to smaller subgraphs, constructing each co-occurrence subgraph from a corresponding page-category subgraph and minimising edge cuts of the category subgraphs. The explanation for the sub-phases are as follows.

4.3.1 Partitioning Page-category Graphs

The page-category graph G^{PC} contains connections of pages and categories where a page is assigned to a category (see Definition 1) as shown in Figure 4.1, and gives details of how the page-category graph is constructed. The graph is stored in a text file of a column list of pages IDs and another list of category IDs. First of all, the category IDs are renumbered to be ascending ordered IDs. Thus, it is easier to operate during the graphs manipulating procedure. The graph is then sorted ascending by page IDs followed by the renumbered category IDs. The edges in the page-category graph are ordered ascending by the page ID, and the edges in the entire graph are unique (no multiple edge).



(a) Subgraph 1 (b) Subgraph 2

Figure 4.4 Two page-category subgraphs

As stated earlier that *t*-component is architected to deal with millions of vertices, it is necessary to partition the initial graph into a set of N subgraphs $\{g_1^{pc}, g_2^{pc}, \ldots, g_N^{pc}\}$. The idea to divide the bipartite graph is simple; The procedure of partitioning the whole graph into smaller pieces ensuring that: (1) each page-category edge $e(g^{pc}) \in E(G^{PC})$ appears in only one subgraph and (2) all subgraphs have approximately the same number of page-category edges. The page ID of the last edge e_i; $p_i \rightarrow c_i$ in each subgraph where the edges count (i) has exceeded the proportion of the total *m* edges and the desired number (N) of subgraphs and a next edge e_{i+1}; $p_j \rightarrow c_j$ must be different. Therefore, the edge e_{i+1} with different page ID will be stored separately in another subgraph. The page-category edges dividing will continue until the whole graph has been partitioned. Note that to speed up the whole

processes while dividing the graph, we can at the same time transform a page-category edge into a category edge on a fly. This will be part of the next process of the graph manipulation in the next section.

An example of dividing the whole graph into two subgraphs ($\mathtt{N} = 2$) can be seen in Figure 4.4, which are partitioned from the entire graph in Figure 4.1. The vertices of the two subgraphs as showed in Figure 4.4 are: $\mathtt{V}(g_1^{pc}) = \{\mathtt{p0}, \mathtt{p1}, \mathtt{p2}, \ldots, \mathtt{p9}\}$ in (a) and $\mathtt{V}(g_2^{pc}) = \{\mathtt{p10}, \mathtt{p11}, \ldots, \mathtt{p20}\}$ in (b). After that the isolated page (following Definition 2) and the isolated category (following Definition 3), which have no meaningful input for the analysis, are removed from the graph; At this point, four pages p0, p18, p19 and p20 were removed, and the current graph remains 17 pages, P = {p1, p2, ..., p17}.

4.3.2 Constructing the Category Co-occurrence Graphs

The investigation focuses on categories. Each page-category subgraph g^{pc} is transformed into its corresponding weighted category subgraph which contains only categories, called a co-occurrence subgraph g^{ew}. The co-occurrence graph G^{EW} contains connectivity of categories where all the pages were removed (see Definition 4). The relationship's strength of the categories is measured by assigning the number of the wiki-pages that they share in common as a 'weight'.

(a) subgraph 1 (b) subgraph 2

Figure 4.5 Two category co-occurrence subgraphs

The procedure of transforming a page-category subgraph into its corresponding co-occurrence graph, follows Definition 4. It has three straightforward processes as follows: (1) constructing a set of categories connected by a same page, (2) creating category edges from the connected categories, and (3) updating weights of the edges as they may be connected by different pages.

First of all, a set of connected categories where categories c_1, c_2, ..., c_n that are linked together by the same page p_i are obtained. Thereafter, all possible unordered category pairs $\{c_i, c_j\}$ are created from the constructed connected categories. To illustrate, in the Figure 4.5 (a) where c1 and c2 are connected with the highest weight of sharing seven pages in common is transformed from the first page-category subgraph in Figure 4.4 (a). While Figure 4.4 (b) presenting the second piece of the page-category subgraph is transformed into the second co-occurrence subgraph as shown in Figure 4.5 (b) where c6 and c7 have four pages in common. The number of pages linking the same two categories together indicating the weight of the category pair has to be subsequently updated. For each page-category subgraph, all category edges are created, their weights are updated, and the weighted edges are stored into their corresponding category graph. To do the transformation for all subgraphs, the procedure must repeat until all of the N page-category subgraphs have been transformed.

4.3.3 Organising Edge Cuts Overlapping Subgraphs

Previously, all page-category subgraphs were transformed into their corresponding category co-occurrence subgraphs. However, there is a fact that the same category vertex may belong to multiple subgraphs. This indicates an *edge cut* which is generally not allow in hard-clustering (non-overlapping members between subgraphs) [284, 285, 287, 288, 302]. For example, within a category subgraph, the same category edge $\{c_i, c_j\}$ may appear more than once, as they can be connected with different pages. Generally speaking,

there could be many different pages shared between the same category pair $\{c_i, c_j\}$ across the different subgraphs. To illustrate this, categories c3, c4 and c5 in the category subgraphs as shown in Figures 4.5 (a) and 4.5 (b), are connected by different pages in bipartite subgraphs (see Figure 4.4). For example,categories c3 and c4 are connected by pages p9 and p10; The pages p8 and p9, and p10 and p11, both pairs connect to categories c3 and c5; Further, categories c4 and c5 are connected by pages p9 and p10.

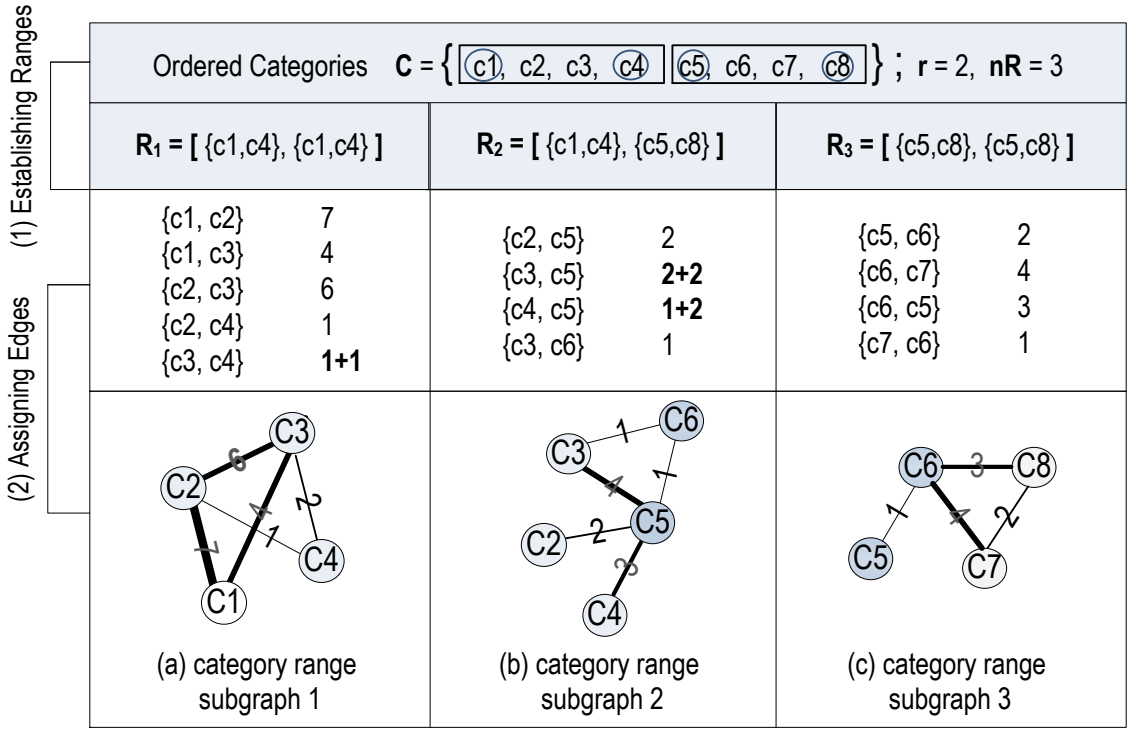


Figure 4.6 Assigning edges into three range category subgraphs

To ensure that the volume of edge cuts is minimised, the edges in the category subgraphs will be assigned into a specific range of edges in a co-occurrence range graph G_R^{EW} (see Definition 6). This guarantees that each category pair belongs to only a single range of edges by redividing the category graph into the category range subgraphs. To achieve that, there are two procedures, **Establishing Ranges** as demonstrated in Algorithm 4 and **Assigning Edges** in Algorithm 5. The outcomes of the two procedures are illustrated in

Figure 4.6, *rows 1-2* of the table presents the first procedure and the algorithm *rows 3-4* presents the second procedure. The two procedures are explained in detail as follows.

(1) Establishing Ranges

Given a set of categories $\mathtt{C} = \mathtt{V}(\mathtt{G}^{\mathtt{EW}}) = \{\mathtt{c}_1, \mathtt{c}_2, \ldots, \mathtt{c}_\mathtt{n}\}$ where $\mathtt{c}_\mathtt{i} \in \mathtt{C}$ from the whole category graph $\mathtt{G}^{\mathtt{EW}}$ and a desired number of subsets $\mathtt{r}$ of the $\mathtt{n}$ ordered categories set $\mathtt{C}$, the possible ranges $\mathtt{nR}$ is $\mathtt{r(r+1)/2}$ ranges. This procedure is to create a set of possible $\mathtt{nR}$ ranges of category pairs $\mathscr{R} = \{\mathtt{R}_1, \mathtt{R}_2, \mathtt{R}_3, \ldots, \mathtt{R}_\mathtt{nR}\}$). A range $\mathtt{R}_\mathtt{i}$ presents a pair of two sets of categories $\mathtt{a}$ for the first unordered category $\mathtt{c}_\mathtt{i}$ and $\mathtt{b}$ for the first unordered category $\mathtt{c}_\mathtt{j}$ of the category edges.

Algorithm 4: Establishing Ranges

Input: $\mathtt{C}$ = a set of ordered categories $\{\mathtt{c}_1, \mathtt{c}_2, \ldots, \mathtt{c}_n\}$; n = number of categories and $\mathtt{r}$ = a desired subsets number of $\mathtt{C}$
Output: $\mathscr{R}$ = a set of $\mathtt{nR}$ ranges of category edges

```
1  a, b, R ← ∅
2  nR = r (r+1)/ 2
3  i = 0
4  (1) Obtaining two sets of categories: a and b (Lines 5-10)
5  while (i < nR) do
6      a ∪ c_{i+=1}
7      b ∪ c_{i+=(n/nR)-1}
8      i ++
9  a ∪ c_{i+=1}
10 b ∪ c_n
11 (2) Constructing R (Lines 12-17)
12 for (j = 1) to nR do
13     R ∪ [{a_j, b_j}, {a_j, b_j}]
14     j++
15     for (k = j+1) to nR do
16         R ∪ [{a_k, b_k}, {a_k, b_k}]
17         k++
18 return R
```

First of all, the two subsets of ordered categories $\mathtt{C}$ are obtained by forming the ranges as shown in Algorithm 4 (*lines 5-10*). Each range $\mathtt{R_i}$ has two subranges: $\mathtt{a}$ and $\mathtt{b}$. Afterward, a possible set of the $\mathtt{nR}$ ranges of the categories is constructed, where each range can be presented as $[\mathtt{a}, \mathtt{b}]$ or $[\{\mathtt{c_{i_a}}, \mathtt{c_{j_a}}\}, \{\mathtt{c_{i_b}}, \mathtt{c_{j_b}}\}]$ for a range $\mathtt{R_i}$. Each range from $\mathtt{nR}$ ranges $\mathtt{R_i}$ of the $\mathscr{R}$ contains two sets of categories $\mathtt{a}$ and $\mathtt{b}$ where each set is a possible combination of the $\mathtt{c_i}$ and $\mathtt{c_j}$ such as $\mathtt{a} = \{\mathtt{c_{i_a}}, \mathtt{c_{j_a}}\}$. This is the constructing $\mathscr{R}$ process as presented in Algorithm 4 (*lines 12-17*). To illustrate, the range $\mathtt{R_2}$, the subset of categories $\mathtt{a}$ is for an ordered pair of $\mathtt{c_i}$ and $\mathtt{c_j}$ (e.g. in between $\mathtt{c1}$ and $\mathtt{c4}$). Also, another subset $\mathtt{b}$ is for an ordered pair of $\mathtt{c_i}$ and $\mathtt{c_j}$ (e.g. in between $\mathtt{c5}$ and $\mathtt{c8}$). While, the $\mathtt{R_1}$ and $\mathtt{R_3}$ have the set $\mathtt{a}$ same as the set $\mathtt{b}$. Note that if the number of the $\mathtt{n}$ categories $|\mathtt{C}|$ is an odd number then the last remaining category will be the last member of the set $\mathtt{b}$.

The outcome of this procedure can be seen in Figure 4.6 (*the first 2 rows in the table*) that given a set of ordered categories $\mathtt{C} = \{\mathtt{c_1}, \mathtt{c_2}, \ldots, \mathtt{c_8}\}$, defined $\mathtt{n}$ to be 2, we have $\mathtt{nR} = 3$. Figure 4.6 (*the first row*) shows the two sets of the categories ($\mathtt{nR} = 2$): $\mathtt{a} = \{\mathtt{c_1}, \mathtt{c_5}\}$ and $\mathtt{b} = \{\mathtt{c_4}, \mathtt{c_8}\}$. These sets are the outcome of Algorithm 4 (*lines 5-10*) for the first process, and Figure 4.6 (*the second row*) are the possible ranges for the second process in Algorithm 4 (*lines 12-17*). The lists of the three $\mathscr{R}$ are $[\{\mathtt{c1}, \mathtt{c4}\}, \{\mathtt{c1}, \mathtt{c4}\}]$, $[\{\mathtt{c1}, \mathtt{c4}\}, \{\mathtt{c5}, \mathtt{c8}\}]$ and $[\{\mathtt{c5}, \mathtt{c8}\}, \{\mathtt{c5}, \mathtt{c8}\}]$.

(2) Assigning Edges

This procedure assigns all edges in the $\mathtt{N}$ co-occurrence subgraphs $\{\mathtt{g}_1^{\mathtt{ew}}, \mathtt{g}_2^{\mathtt{ew}}, \ldots, \mathtt{g}_{\mathtt{N}}^{\mathtt{ew}}\}$ into the previous constructed category range form new category range subgraphs $\{\mathtt{g}_{\mathtt{r}_1}^{\mathtt{ew}}, \mathtt{g}_{\mathtt{r}_2}^{\mathtt{ew}}, \ldots, \mathtt{g}_{\mathtt{n_R}}^{\mathtt{ew}}\}$, where each range stands for the new subgraph. The edges of $\mathtt{c_i} \rightarrow^{\mathtt{w}} \mathtt{c_j}$ sharing different pages across the subgraphs are assigned into a specific range of edges $[\mathtt{a}, \mathtt{b}]$ of the $\mathscr{R}$ where $\mathtt{c_i}$ must belong to $\mathtt{a}$ and $\mathtt{c_j}$ belong to $\mathtt{b}$. Doing this can ensure that all the edges are unique.

Algorithm 5: Assigning Edges

Input: $\mathtt{N}$ category co-occurrence subgraphs $\mathtt{G^{EW}} = \{\mathtt{g_n^{ew}} | \mathtt{n} \in \mathtt{N}\}$ and
$\mathscr{R}$ of weighted-edges category co-occurrence ranges

Output: $\mathscr{R}$ category range graphs $\mathtt{G_R^{EW}} = \{\mathtt{g_{R_i}^{ew}} | \mathtt{R_i} \in \mathscr{R}\}$

```
1  repeat
2      reading each weighted category edge c_i →^w c_j in the g_r^ew
3      for (each range of ℛ) do
4          if (c_i ∈ a ) and (c_j ∈ b) then
5              if (c_i →^w c_j ∉ g_r^ew) then
6                  w=w
7              else
8                  w+=w
9      g_n^ew ∪ c_i →^w c_j
10 until end of G^EW;
11 return G_N^EW
```

Algorithm 5 shows the assignment procedure. If the new assigned edge has already appeared in the new range graph then its weight must be summed (*lines 7-10*-Algorithm 5). For each subgraph, all category pairs are read and assigned into their corresponding range of edges, and the weights of the edges are updated, after that the weighted category edges are stored into a new category range subgraph. This procedure is repeated for all subgraphs. To illustrate, the three category range subgraphs shown in Figure 4.6 (*rows 3-4*) are obtained from the two input category subgraphs in Figure 4.5 (a) and (b). The edge `{c3,c4}` is assigned to the edge ranges `[{c1,c4},{c1,c4}]`, the weighted category range subgraph `1`, because `c3` and `c4` belong to the range `{c1,c4}`. For this edge in the new range graph, the weight has been summed up to be `2`. The process will continue until all `N` category subgraphs have been reformed into the `nR` category range subgraphs $\mathtt{G_R^{EW}}$.

4.4 Graph Clustering

The second phase in the framework is a procedure to perform structural clustering. This phase has three tasks: (1) filtering subgraphs by increasing a weight threshold (2) identifying category clusters within each subgraph, and (3) combining clusters identified from all subgraphs as shown in Figure 4.3.

M-core (explained in Chapter 2) is used to obtain nested subclusters. An important goal is to analyse the clusters by investigating how the clusters's properties are related to the weight threshold. For each core[1], BFS (as shown in Algorithm 1-Chapter 2) is used to identify the clusters within the filtered graph. It must be emphasised that the t-component framework is designed to deal with a large graph, where the entire graph is too large to be held in memory, so that processing smaller subgraphs is the only option. The resulting clusters for the whole graph is a combined cluster from the subgraphs. To avoid confusion of terminology for these processes and their outputs, a parameter t is utilised for the threshold of the weighted category edges, the output of a subgraph filtered from a weighted range category subgraph is called t-filtered category graph (see Definition 7), and at each weight threshold t corresponding to a current threshold t that all edges have weight equal or greater than t.

4.4.1 Filtering Subgraphs

In the previous phases, the $\mathtt{R}$ range category graphs have been obtained. This stage is to obtain $\mathtt{R}$ t-filtered category graphs $\mathtt{G}^{\mathtt{EW}}_{\mathtt{R},t}$, which each t-filtered graph corresponds to category range graph where the weighted category edges exceed t value.

[1] recall that a 'core' contains maximal (highest amount of connected categories) subgraphs of an entire graph (detailed in 2.5.1)

The filtering subgraphs is performed for every t-filtered category graph obtained by retaining just a single edge linking each pair of categories when the weight of the edge exceeds the specified threshold t. The analysis in this thesis sets the minimum weight at 2 for the filtering category graph, excluding the feeble category edges (see Definition 5). Feeble edges having weak relation, i.e. each category pair has only one page in common. The highest weight of the category edges can be found by seeking the edges' max weight (sorting edges descendingly by their weight). To obtain the filtered category graph $\mathtt{G}^{\mathtt{EW}}_{t}$, for each $\mathtt{G}^{\mathtt{EW}}_{\mathtt{a,b}} \subset \mathtt{G}^{\mathtt{EW}}$, all edges with weight less than t are removed. Hence, each $\mathtt{G}^{\mathtt{EW}_t}_{\mathtt{a,b}}$ is converted to its corresponding t-filtered category graph $\mathtt{G}^{\mathtt{EW}}_{\mathtt{a,b,t}}$. It is easy to see that $\mathtt{G}^{\mathtt{EW}}_{t} = \bigcup_{\mathtt{a,b}\in\mathscr{R}} \mathtt{G}^{\mathtt{EW}}_{\mathtt{a,b},t}$.

Algorithm 6: Graph Clustering based on m-core nested clustering

Input: range category graph $\mathtt{G}^{\mathtt{EW}}_{\mathtt{R}}$

Output: combined category cluster $\mathtt{CC}$ $(\mathtt{G}^{\mathtt{EW}}_{\mathtt{R},t})$

```
1  for (each t-filtered subgraph) do
2      (1) Filtering subgraphs
3      while (w(g^e_{r,t}) ≥ t) do
4          g^{ew}_{r,t} ∪ e(g^w_r)
5          (2) Identifying category clusters using Algorithm 1: BFS
6          (3) Combining clusters
7          for (each cluster C_i of G^{EW}_{R,t}) from (i=1 to |C_i|) do
8              if a category c in C_{i+1} ∈ C_i then
9                  C_i ∪ c
10         return cc(g^{ew}_{r,t}) ∪ c(g^{ew}_r)
11 return combined category cluster CC (G^{EW}_{R,t})
```

Algorithm 6 (*lines 3-4*) performs the filtering subgraphs procedure by obtaining $\mathtt{R}$ t-filtered category graphs corresponding to a core where the weight threshold t starts from 2. All edges of the co-occurrence category graph $\mathtt{G}^{\mathtt{EW}}$ with weight equal to or greater than t are

retrieved; During this phase all feeble category edges (weight with 1 see Definition 5) are ignored. Note that the filtering process can be executed individually from a certain threshold range. For a better performance of the procedure, a current t of a filtered graph is the starting category graph for a next (t+1) filtered graph. Figure 4.7 shows the three t-filtered graphs (t from 2 to 3) filtered from the range subgraphs in Figure 4.6.

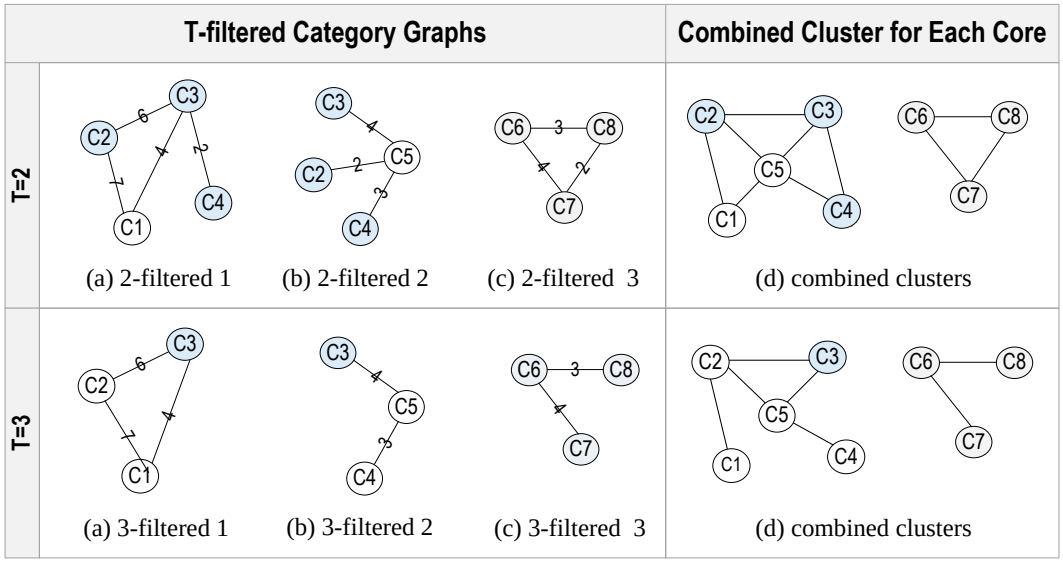


Figure 4.7 t-filtered subgraphs, clusters and combined clusters

4.4.2 Identifying Category Clusters

This procedure aims to identify category clusters as the connected component within each t-filtered category graph corresponding to each core for a specific t. The clusters $\mathtt{C}(\mathtt{G}^{\mathtt{EW}}_{\mathtt{R},t})$ within each $\mathtt{G}^{\mathtt{EW}}_{\mathtt{R}} \subset \mathtt{G}^{\mathtt{EW}}$ are identified by using BFS (see Chapter 2-Algorithm 1).

Algorithm 6, shows the search for clusters for each t-filtered category graph at a specific core with a specific weight-threshold t. Note that to complete for all subgraphs, this must run every subgraph in turn. The process can be executed individually from a range of thresholds t as a starting threshold to t+n as an end threshold. Figure 4.7 presents three $t2$-filtered

subgraphs and another three *t3*-filtered subgraphs where each one of these subgraph examples represents a cluster. However, the real graph for each filter graph would contain many clusters.

4.4.3 Combining the Clusters

As stated earlier, to handle the large graph, it has been necessary to partition it into smaller subgraphs where each subgraph is operated in turn. Previously, the category clusters corresponding to their *t*-filtered category subgraphs have been obtained and each one is stored in a separate text file. However, there may be at least a category c_i that links clusters in different subgraphs together. These clusters have to be merged into a combined category cluster, following Definition 9.

This phase performs the process to combine all clusters across the subgraphs by merging any category clusters that share at least one category in common. To perform the merge, the first category cluster $C(g^{ew}_{r,t})$ is initially assigned as the resulting combined cluster $CC(G^{EW}_{R,t})$. The merge begins by taking the next cluster $C(g^{ew}_{r+1,t})$ where $r+1$ starts from 2 to R. Every cluster within the first (or current) category cluster $C(g^{ew}_{r,t})$ to every cluster within the next cluster $C(g^{ew}_{r+1,t})$ are merged if any pair of clusters share at least a category in common. Then the merge will continue until all the next category cluster $C(g^{ew}_{r+1,t})$ where $r+1 = R$ (last cluster) have been merged. The complete set of category clusters for threshold *t* as a combined cluster of $CC(G^{EW}_{R,t})$ are obtained. In other words, the resulting clusters are combined by connecting pairs of clusters that share at least one category into a singe category graph $CC(G^{EW}_{R,t})$ corresponding to a *t*-filtered category graph. This process is repeated until all clusters are combined.

Figure 4.7 demonstrates the obtained subgraphs where t is 2 and 3. To illustrate the clusters for the subgraphs when t is 2, the three category clusters: `cluster-subgraph1` = {`c1,c2,c3,c4`}, `cluster-subgraph2` = {`c2,c3,c4,c5`}, and `cluster-subgraph3` = {`c6,c7,c8`}. For the first pair of subgraphs, the two clusters in `subgraph1` and `subgraph2` were compared to see whether they have a category in common, and the result of the combined cluster is {`c1,c2,c3,c4,c5`}, this is the new current combined cluster. The next cluster subgraphs pair is compared now, and after this merge has been performed, the resulting cluster is {`c6,c7,c8`} where all categories from the three cluster subgraphs are combined. There are the same resulting clusters for the two, at setting threshold t.

At this stage a collection of different sizes of category clusters in the corresponding t-filtered category graph is obtained. The graph properties to be obtained are the number of categories and clusters, the number of categories in the largest cluster, and the number of the smallest clusters (cluster size 2) in each t-filtered category graph corresponding to each core.

4.5 Chapter Summary

The major goals of the investigation are clustering and analysing the structural clusters. This requires a methodology to examine the interactions among the pages and categories and categories connectivity in Wikipedia. The t-component framework has been introduced which provides the analytic methodologies that can handle a large graph. The framework has functionalities to perform exploring, manipulating and clustering of the graphs. The structural properties of the graph and the cluster sizes for different weight thresholds are observed and analysed.

First, the graph manipulation phase divides a large bipartite graph into smaller bipartite subgraphs. Each of subgraph is then transformed into its corresponding weighted category subgraph which contains only categories. The categories connectivity or strength of relationships are measured by assigning the number of the wiki-pages that they share in common as a weight. Further, the edges of the weighted category subgraphs are assigned into unique edge ranges. This is to ensure that there are only edge-disjoints within the whole weighted category graph.

Second, the core clustering has been introduced which used the BFS to identify category clusters. The core structural clustering is based on the *m*-core method, the nested cores filtering subgraphs which concerns multiplicity of the weighed edges. Afterward, all clusters across the subgraphs that share at least one category in common will be merged into a single cluster for each core. This cluster corresponds to a core of *t*-filtered category graph. Finally, the clusters result corresponding to the core can be analysed in relationship to the clusters structural properties and the *t*, weights threshold.

The methodology contributions of the *t-component* framework are (1) use the graph partitioning technique to handle the large bipartite graph (2) transform them into a weighted subgraph (3) ensure all weighted-edges are unique (4) filter the subgraphs based on *m*-core to retrieve only edges having weight at a certain value (5) identify category clusters and (6) combine the clusters.

The proposed analytic framework is used to manipulate the multiple large bipartite graphs representing Wikipedia page-category link networks and clustering the categories within the co-occurrence graphs. The next chapter will present and discuss the findings, and also the results will be validated and represented.

Chapter 5

Graph Clustering Results

The t-component framework, introduced in Chapter 4, was used for clustering categories on the multiple Wikipedia category co-occurrence networks. The content in this chapter presents the clustering results and insights of the analysis. The results on structural properties of the page-category graphs and category graphs clustering are presented in the first two sections. A comparison of clustering results using different cohesive approaches such as the k-core and m-core are shown and discussed in Section 3. Insights into the separation of category hubs are discussed in Section 4. The final section is the chapter summary.

5.1 Results on Graph Properties

This section presents the evolution statistics of Wikipedia networks not only for English editions but also the German, the second largest edition after the English. The studies [13, 73, 74] revealed the structural properties of both editions, however the information was for the early time of Wikipedia, over 10 years ago.

Network Properties	English-10	English-11	English-12	English-15	German-10	German-11	German-12
#pages	8,989,264	12,182,689	12,453,596	18,186,169	1,654,787	1,869,899	2,007,395
#categories	567,939	801,902	858,869	1,325,069	94,724	130,866	146,106
#page-category edges	39,484,287	56,969,309	60,386,600	94,134,597	4,946,017	5,828,101	6,573,688
#weighted category edges	7,450,892	11,640,934	12,788,867	22,348,527	2,676,044	3,357,499	3,874,033
#isolated pages	1,083,655	1,735,857	1,755,160	3,132,601	574,464	571,555	598,202
#isolated categories	7,443	7,858	7,375	8,877	1,739	2,000	2,072
%isolated pages	12.05%	14.25%	14.09%	17.23%	34.72%	30.57%	29.80%
%isolated categories	1.31%	0.98%	0.86%	0.67%	1.84%	1.53%	1.42%

The rows of network properties are:**#pages** = total number of unique pages in the page-category networks; **#categories** = total number of unique categories in the the page-category networks; **#page-category edges** = total number of the links from pages connecting to categories in the page-category network; **#weighted category edges** = total number of category links in category co-occurrence network where frequency of the shared pages for each category edge is concerned as the edge's weight (excluding feeble edges having weight = 1); **#isolated pages** = total number of pages connecting to at most one category; **#isolated categories** = total number of categories not sharing any pages with any other category; **%isolated pages** = percentage of isolated pages; **%isolated categories** = percentage of isolated categories. There are English and Germen networks presented in the seven columns such as **English-10** shortened for Wikipedia edition English year 2010.

Table 5.1 Summary of evolution statistics of Wikipedia English and German networks

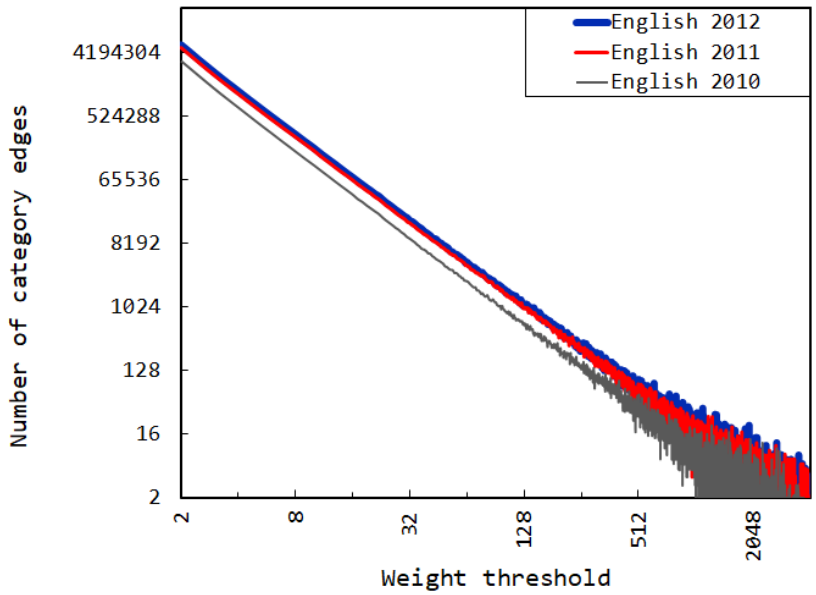


Figure 5.1 Log-log plots-the number of category edges for different weight threshold values for English category co-occurrence networks 2010-2012

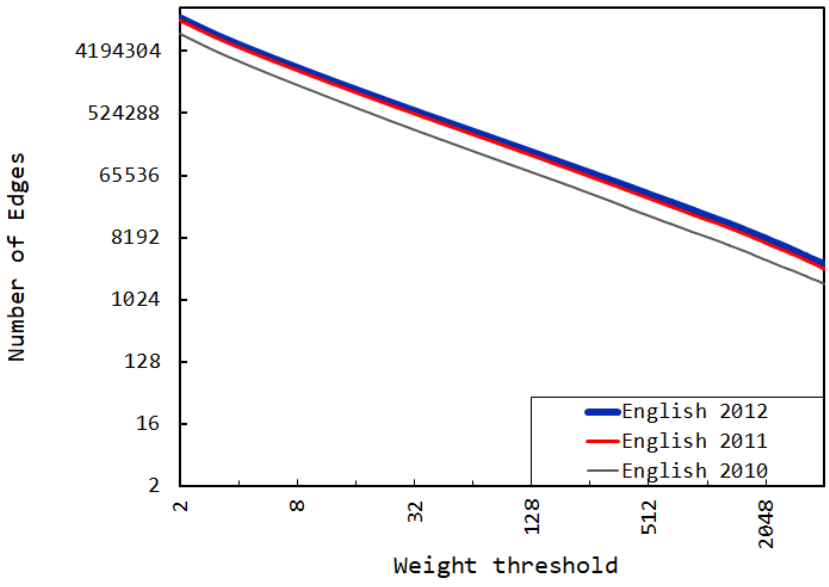


Figure 5.2 Log-log plots-the cumulative number of category edges for different weight threshold values for English category co-occurrence networks 2010-2012

Basic statistics of the nontrivial networks excluding the feeble edges having weight 1 are summarised in Table 5.1. The individually discussed language editions are compared.

First, in the English networks between 2010 and 2015, the number of the pages and categories were increased by around 40% and 50%. It is interesting that over time, the growth of categories was higher than the pages. Likewise, the results of [66] from 2012 to 2014 for the growth of categories and pages are 25% and 12%. It can also be observed that the number of edges of the page-category networks and category co-occurrence networks increased by around 50% and 70%. A consequence of this, which can be checked using Table 5.1, is that the average degrees, both pages per category and categories per page, were unchanged. The average degrees of the page-category (about 9) and category networks (about 30) were also unchanged during the years observed. It was reported that a large volume of admin-categories were linked to articles by approximately 70% [66]. Also, in recent years, it is possible that there may be more administrative categories established to maintain the (article) pages. We shall return to this point in Section 4, where the proof for English edition 2015 is presented. Noticeably, as far as where the edges growth is high, there are a great number of isolates and also feeble category edges, which are connected by only the same page (with weight 1). For example, there is almost 70% (75,059,266) feeble edges without isolates of the total category edges for the network edition 2015, and on average 60% for the other four English editions. An interesting observation is that the number of isolated categories was almost unchanged, even though the number of isolated pages increased by approximately 60%. An explanation for this is that most new categories are possibly assigned into existing categories like the result was interpreted in [66].

The relationship between the number of category edges and the weight threshold for all values of ranges from 2 to 4096 for the three English networks in log-log scale is presented in Figure 5.1 exhibiting heavy tails. It shows that categories sharing an extremely high number of pages in common rarely appear. While, those categories sharing fewer pages tend

to appear regularly. When a cumulative distribution of the weight threshold is plotted in Figure 5.2, it shows a straight line with a declining slope around 1 on a linear regression. This indicates a power-law determining a scale-free network as explained in Chapter 2, Section 2.2. Their relationship suggests power-laws like those studies in different properties in Wikipedia networks [13, 73, 74] and in networks of Wikipedia categories [53, 66, 180].

Next, the summarised structural properties of the three German networks over the years 2010 to 2012 can be reviewed in Table 5.1. The number of the pages and categories have increased by around 20% and 50%. The observed number of page-category edges and weighted category edges increased by approximately 30% and 40%. The average degrees of the page-category and category networks, between 6 and 50 degrees, were also unchanged during the observed years. It is interesting to note that the growth of categories is about two times higher than the pages, and the number of isolated pages was almost unchanged. However, the number of isolated categories increased by roughly 20%. A possible explanation for this (correspondingly in [66]) is that the relationship of a page assigned to multiple categories may be removed, and only the most related content category remains. Consequently, it has left these categories isolated. Furthermore, the relationship between the number of category edges and the weight threshold is presented by two figures in Appendix B: A pair of network properties, are plotted in log-log scale as shown in Figure B.1; Their cumulative plots in log-log scale can be seen in Figure B.2, fitted well by a linear regression line with exponent value approximately 1.40 (approx $R^2 = 0.98$) for the three editions. It can be seen that all networks plotted in the two charts follow power-laws when the amount of shared pages vary.

Finally, using Table 5.1 to compare the network properties in the two editions, it is noticeable that the English network is almost ten times larger than the German network for each year observed. Overall, there is not much difference in the structural properties between the two languages such as the growth of the pages, categories, page-category and category

edges over the years observed. The English editions have a higher growth of the observed network properties (about 60%) than the German (about 40%). Also, the growth of categories is higher than the pages in both English and German networks. It is very interesting to note that the number of English isolated pages increased substantially by roughly 60%, while the number of the isolated categories is almost unchanged over the period of time observed. In contrast, the German isolated pages is almost unchanged, but the isolated categories increased by about 20%. An explanation is that the new wiki-pages contributed into the English category network were more likely to be assigned into related existing categories (large domain topics) following the Wikipedia guidelines. The English categories would probably be well maintained as a self-organised system (usually has scale-free characteristic) [284]. It is possible that the German pages would also be categorised into multiple non popular categories.

5.2 Results on the Graph Clustering

After performing m-core clustering on the co-occurrence networks, the category clusters were identified. A few clusters's properties are analysed such as the number of clusters, the number of cluster size two, number of categories, and sizes of the largest cluster. The following findings in the clusters' structure were all observed values of weight threshold t from 2 to 4096 in the English co-occurrence networks over the years 2010 to 2012 and 2015. Each of the resulting charts presented in the following pages are plotted in the log-log scale in order to linearise a power-law relationship between the observed properties versus the weight threshold. Note that the clustering was also performed on the German editions where the results can be seen in Appendix B, Figure B.3 to Figure B.6, compared with the English editions. As the results show there were no differences between the two editions that all observed clusters's properties following the power-law distribution with respect to the threshold t.

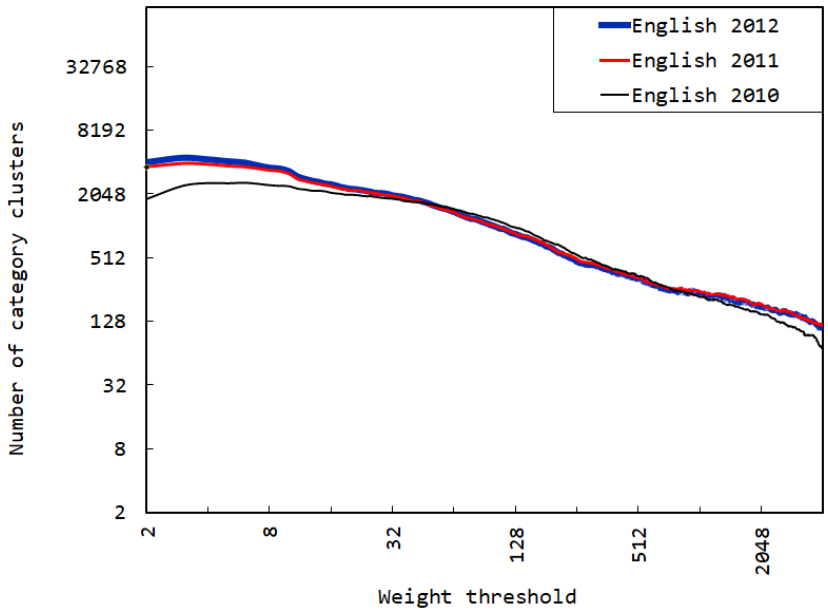


Figure 5.3 Log-log plots-the number of category clusters for different weight threshold values for English category co-occurrence networks 2010-2012

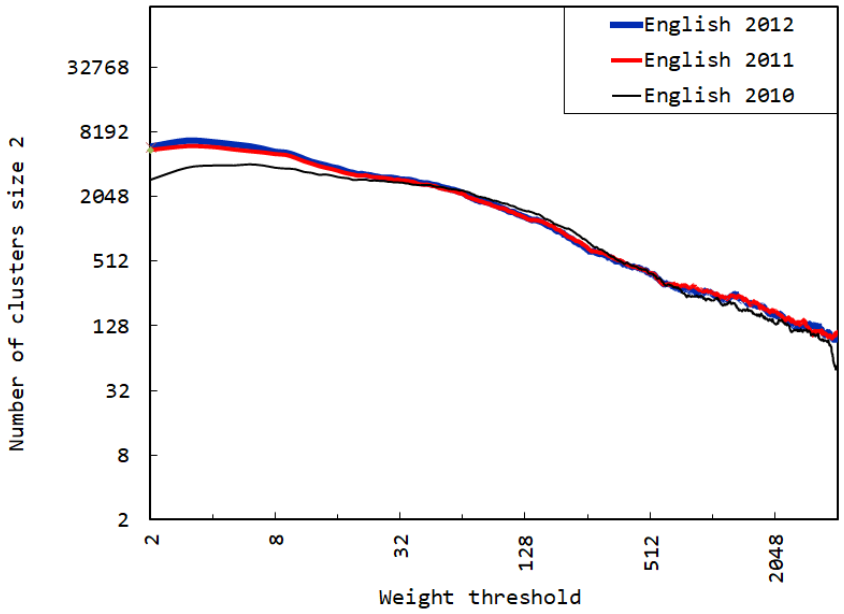


Figure 5.4 Log-log plots-the number of cluster size two for different weight threshold values for English category co-occurrence networks 2010-2012

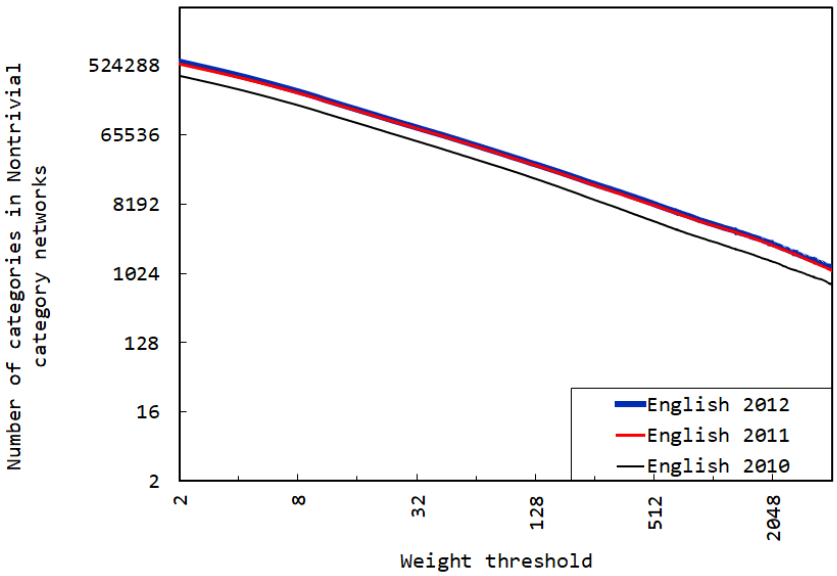


Figure 5.5 Log-log plots-the number of categories for different weight threshold values for English category co-occurrence networks 2010-2012

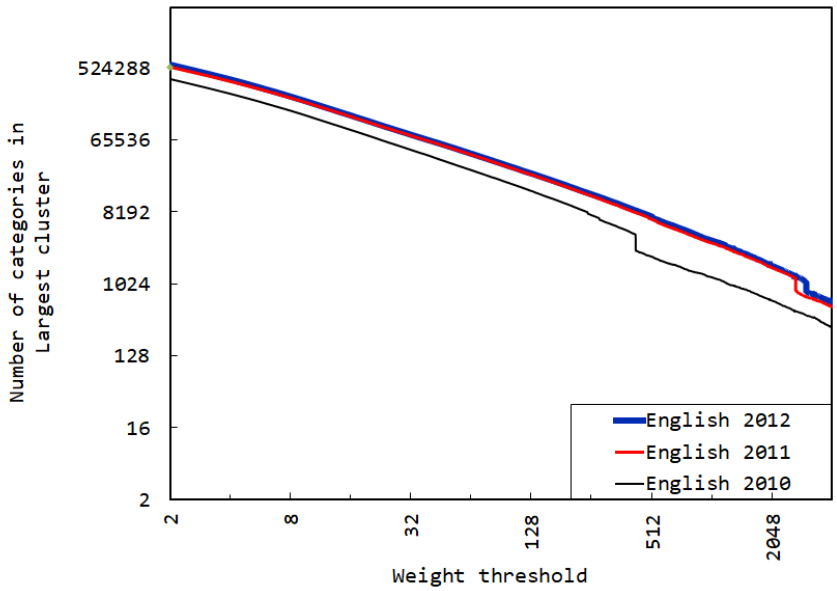


Figure 5.6 Log-log plots-the size of the largest cluster for different weight threshold values for English category co-occurrence networks 2010-2012

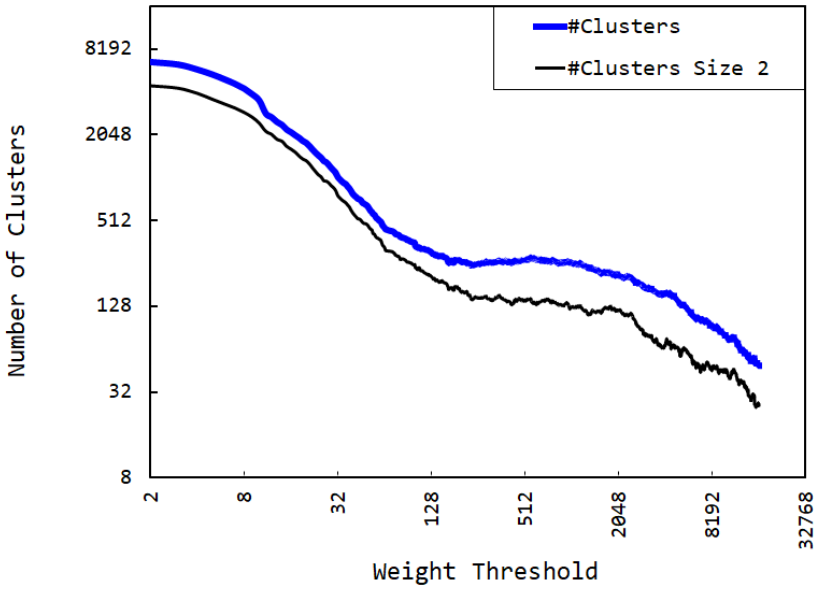


Figure 5.7 Log-log plots-the number of category clusters and the cluster size two for different weight threshold values for English co-occurrence network 2015

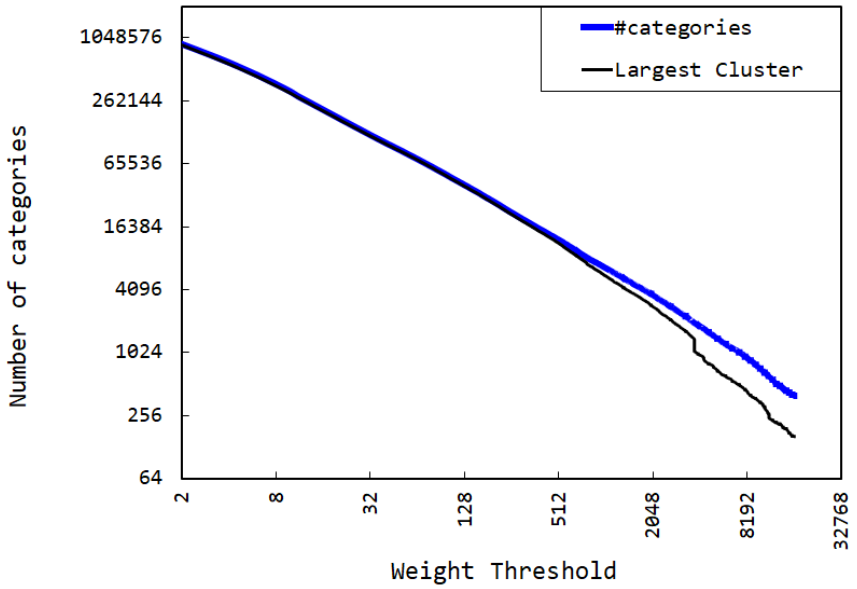


Figure 5.8 Log-log plots-the number of categories and the size of the largest cluster for different weight threshold values for English co-occurrence network 2015

This section only discusses the four English editions, which have the more significant growth and changes in sizes of the largest category clusters than the German's. The number of clusters are plotted against the weight threshold in the log-log scale for each of the four English editions, which show almost the same straight line as presented in Figures 5.3 for 2010-2012 and in Figure 5.7 for 2015. The log-log plots in Figure 5.4 presents the relationship between the number of smallest clusters, containing only two categories for the editions 2010-2012, and Figure 5.7 for 2015. It can also be seen that all the log-log plots are almost the same straight line. The observation is that the majority of clusters in the category network contains only a few categories, mostly two. This shows significantly that for every weight threshold, the majority of the clusters are small with a size of two.

The log-log plots of the number of categories in the networks 2010 to 2012 are presented in Figure 5.5, and the largest cluster can be seen in Figure 5.6. The two properties, number of categories and size of the largest cluster for network 2015 are plotted in log scale on the same chart as shown in Figure 5.8. Interestingly, for each of the four networks, the size of the largest cluster (*giant cluster*) has dropped sharply at a specific weight threshold indicating *giant cluster split*. For instance, it is observed in the 2010 network that there are $1,601$ categories in the largest cluster, which dropped significantly at the threshold $t = 427$. This suggests the phenomenon of a hub with 'rich get richer' behaviour indicated by where the giant cluster occurs; The evidence shows the large category cluster has fallen apart after increasing the threshold t from 426 to be 427. There is a threshold value for every network where the size of the largest category cluster drops sharply; Each large category for each of the four editions has divided into smaller categories. To validate this, the appearance of the category hubs will be tested on the random graphs, demonstrated in Chapter 6, Section 6.2. Overall, a very significant finding here is that all observed properties represented in the charts of the log-log plots appear to exhibit the power-law behaviour with respect to the threshold t.

Co-occurrence Networks	#Categories		Sizes of Largest Clusters	
	α	R^2	α	R^2
English 2010	-0.891	0.9991	-1.075	0.9913
English 2011	-0.889	0.9968	-1.088	0.9690
English 2012	-0.881	0.9979	-1.048	0.9767
English 2015	-0.986	0.9935	-1.223	0.9856
German 2010	-1.340	0.9690	-1.374	0.9363
German 2011	-1.211	0.9680	-1.299	0.9379
German 2012	-1.246	0.9570	-1.272	0.9393
French 2012	-0.882	0.9990	-1.027	0.9966
Russian 2012	-1.056	0.9872	-1.265	0.9755
Italian 2012	-1.183	0.9850	-1.345	0.9792
Portuguese 2012	-0.814	0.9950	-0.902	0.9793

Columns from left to right are: **Category Networks** = nontrivial category co-occurrence networks without isolated categories; **#Categories** = amount of categories in the category co-occurrence networks; **Size of Largest Clusters** = amount of category vertices in the largest cluster; α = power-law exponent value, decline slope; R^2 = coefficient of determination value denoting how well of the fitted linear regression line and all of the data points, $0 \leq R^2 \leq 1$.

Table 5.2 Comparison of power-law exponents and goodness fit for multiple category co-occurrence networks

The studies [73, 74] revealed the strongly connected components, a part of the bow-tie components of the Wikipedia networks for many languages that follow the power-laws distribution. However, the analyses were focusing on the connected pages components in the Wikipedia page-links networks, but the relationship of categories were not their concern. Indeed, this thesis' analysis focuses on the splitting phenomena of the giant category clusters, not only for the English editions but also for multiple languages.

The power-law exponent (α) and coefficient of determination values (R^2) for 11 editions in the six language editions can be compared using Table 5.2. It can be seen that the power-law exponent of the English versions is not significantly changed over the time period examined. It is also noticeable that these power-law exponent values are all below the expected range (between 2 and 3) in real world networks [85, 96]. This is similar to the findings reported in the last section that increasing the shared pages threshold caused

a decline in the number of weighted edges. An explanation is that for each network, for every threshold t, the majority of the clusters are small with size of two. There were only a few large clusters which contained most of the categories in the co-occurrence network. This suggests an existence of a few hubs in a scale-free network as found in the other networks [7–9, 92–95]. Note that the m-core clustering results where the giant cluster has split is validated on random graphs and a taxonomy graph in Chapter 6. However, one wonders if this would appear for the popular k-core's, the next section has the intension to experiment on this.

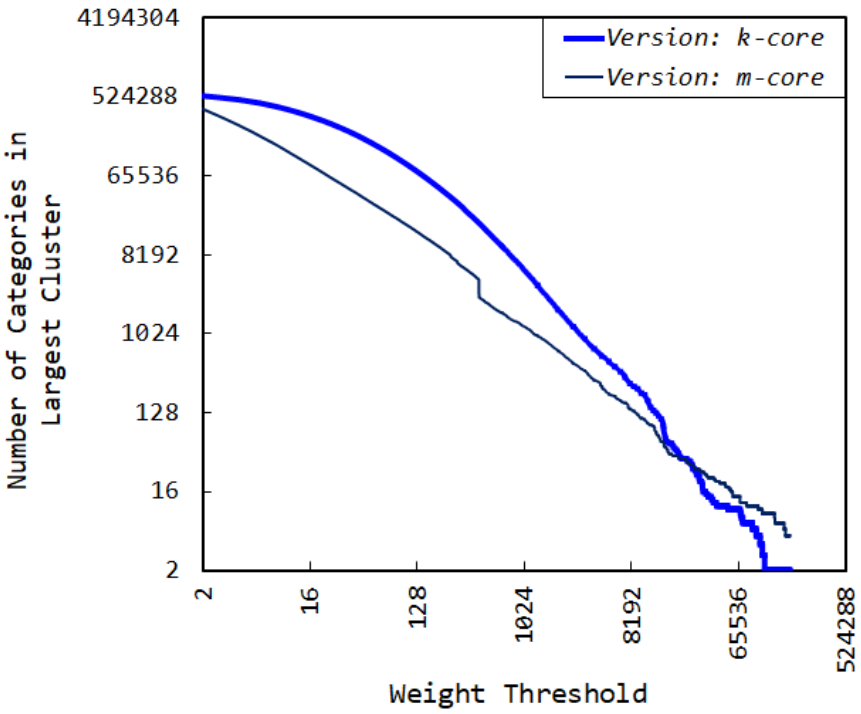


Figure 5.9 Log-log plots-comparison sizes of the largest clusters of m-core and k-core for English category co-occurrence network 2010

5.3 Results on k-core Versus m-core

The first two chapters explained the difference between the the m-core and k-core clustering models in terms of cohesive measurement; The category relationship for the m-core is concerned with edge frequencies or the amount of pages they share. Whilst, the k-core uses the number of vertex' neighbours or the category's degree to quantify the categories cohesion. The threshold of the shared page frequencies is for the m-core, and the k-core uses the threshold of category's degree.

The m-core is used as the constituent component to group categories in the t-component framework as presented in the previous chapter. In this section, the English co-occurrence network 2010 was performed in clustering by employing the k-core and the result of the category relationship is compared to the m-core's. The chart in Figure 5.9, on next page presents the category clustering results for the largest clusters' sizes obtained by using the m-core and k-core plotted in the exponential scale. It can be seen that increasing the weight threshold values of both models caused the entire categories in the network to be more disconnected.

There are a few differences between these two. It is significant that the volume of categories of the k-core version are remarkably higher than the m-core's. Also, its declining ratio in size of the largest connected categories is larger than the m-core's. The slope is roughly twice that of the m-core's ($\alpha \simeq$-1.60 for the k-core and $\alpha \simeq$ -0.80 for the m-core). The most crucial result shows that only the m-core version exhibits the giant split phenomenon. The insights of what causes the categories to be separated, will be presented in the next section.

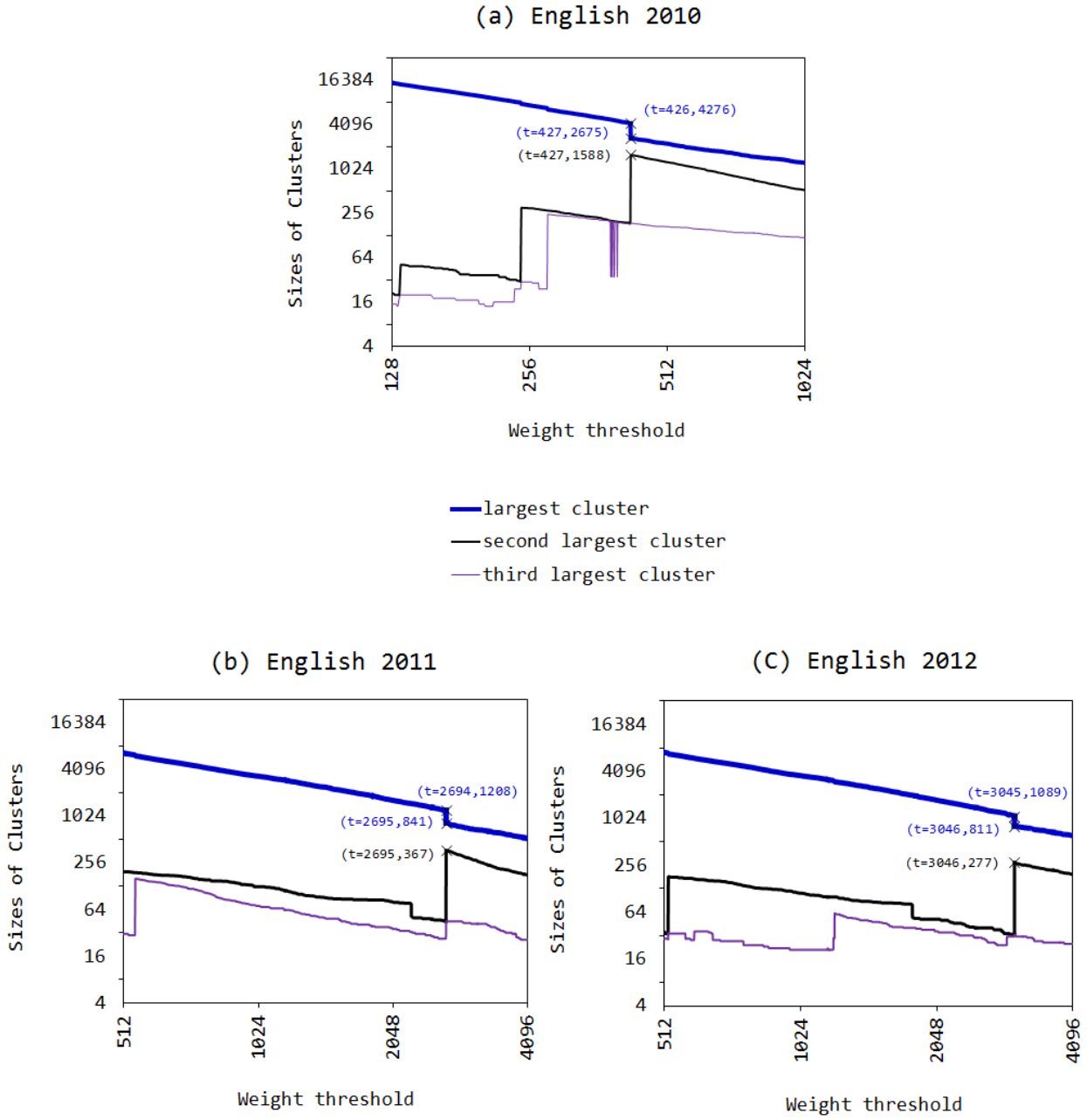


Figure 5.10 Log-log plots-the three significant points of the largest clusters separating into two large category clusters for the three English Wikipedia category co-occurrence networks from 2010 (a) to 2012 (c)

5.4 Insights of Wikipedia Category Hubs

Previously, in all English editions observed, the largest cluster divided into two smaller clusters were found at the critical weight threshold between t and t+1 as presented in Figure 5.6 and Figure 5.8. This section reveals the insights of the category hubs separation.

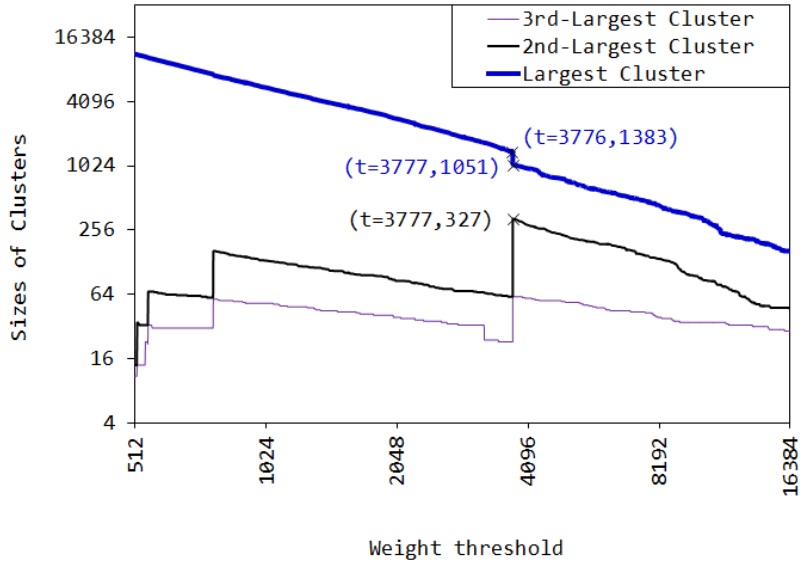


Figure 5.11 Log-log plots-the three significant points of largest clusters separation for the English Wikipedia category co-occurrence network 2015

Taking a closer look at the critical threshold interval from t to t+1, Figure 5.10 shows the three largest clusters plotted in log scale against the weight threshold for the editions 2010 to 2012. It can be observed that in each of the editions from charts (a) to (c), the largest cluster is separated significantly into two large subclusters. It has the significant points of the clusters separation in the corresponding sizes of the first and second-largest clusters. Charts (b) and (c) for the 2011 and 2012 editions look alike, when increasing the share pages threshold, there are about 70 and 20 percent of categories in the largest and second-largest clusters. Their proportion of the share pages threshold and the cluster sizes are quite similar.

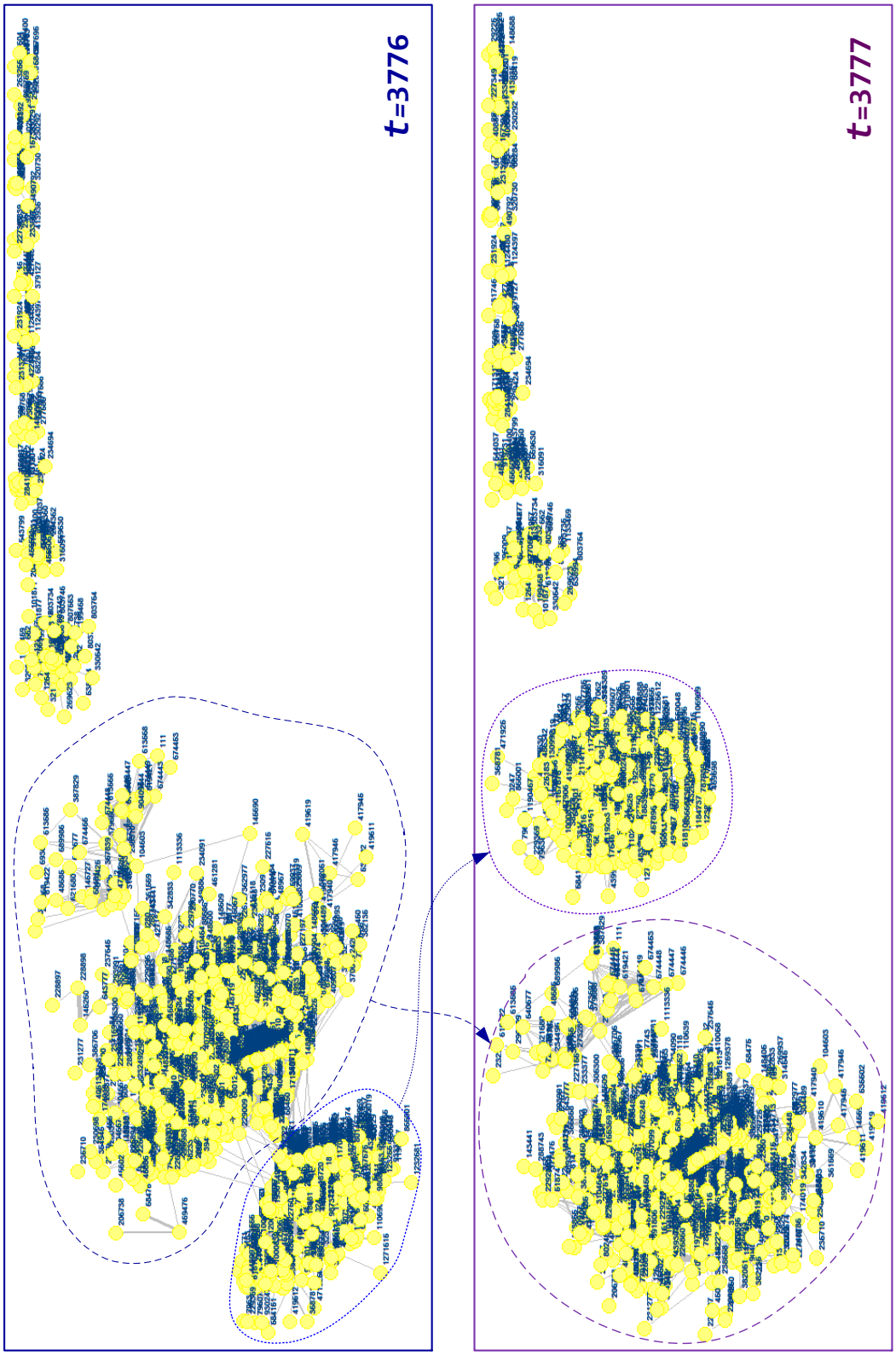


Figure 5.12 Visualisation of the significant categories splitting of the largest category cluster at weight threshold 3776 into two smaller clusters at threshold 3777 for English Wikipedia category co-occurrence networks 2015

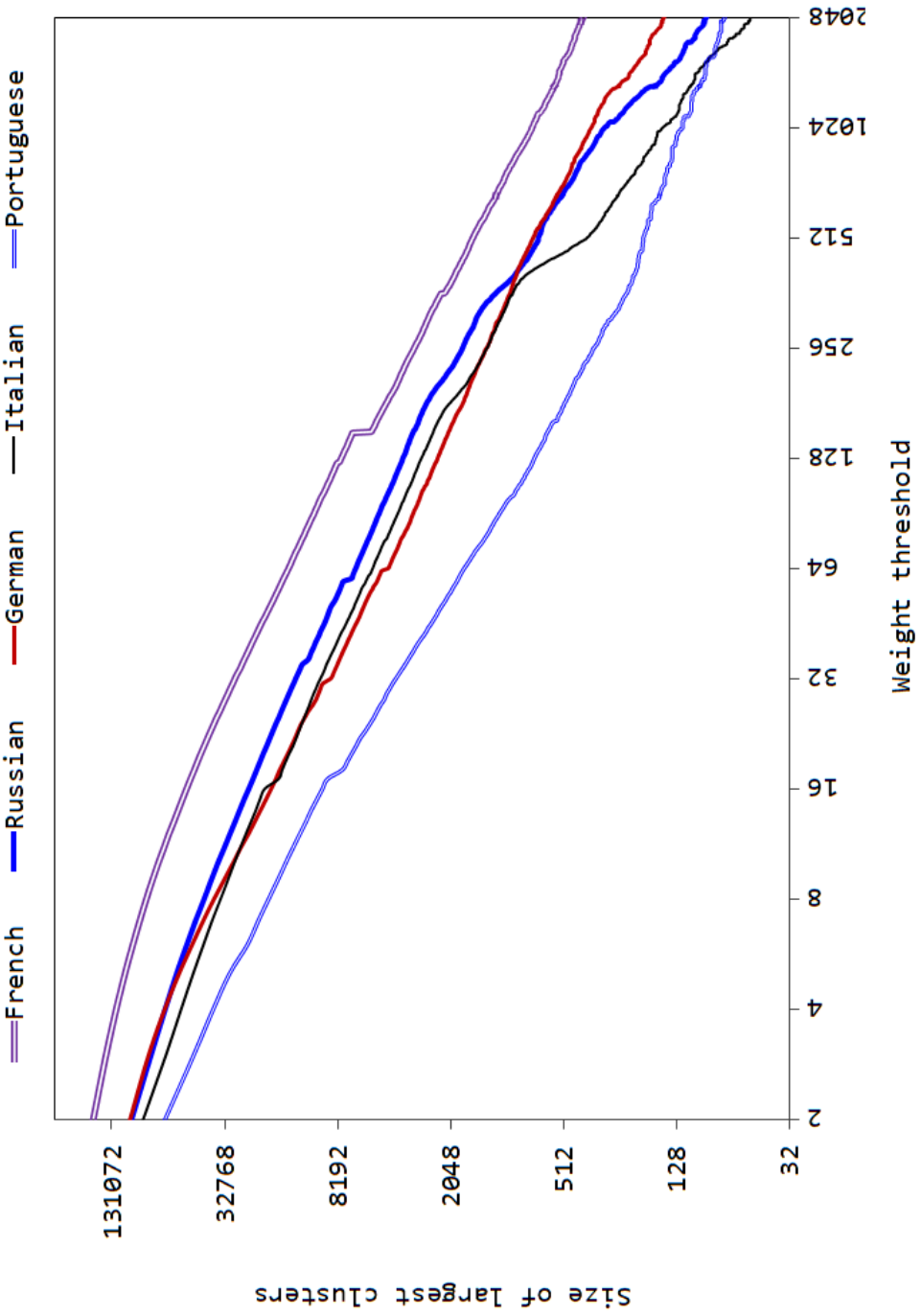


Figure 5.13 Log-log plots-number of categories in the largest cluster for all values of weight threshold from 2 to 2048 for the five languages of Wikipedia category co-occurrence networks 2012

Networks	#Category Edges	#Categories	Size C1	*t*1:Size C1	*t*2:Size C1	%Size C1-Dropped
English 2010	3,181,496	387,689	383,262	*426* :4,276	*427* :2,675	37.44%
English 2011	4,896,880	552,535	542,402	*2694* :1,208	*2695* :841	30.38%
English 2012	5,373,630	591,474	580,255	*3045* :1,089	*3046* :811	25.52%
English 2015	9,239,386	915,564	896,034	*3776* :1,383	*3777* :1,051	23.73%
French 2012	1,274,086	164,838	164,581	*150* :6,838	*151* :5,427	20.63%
German 2012	1,193,484	106,903	104,683	*31* :9,842	*32* :8,929	9.28%
Russian 2012	1,844,325	104,425	102,740	*59* :7,624	*60* :6,924	9.18%
Italian 2012	1,280,033	93,996	89,178	*16* :20,067	*17* :16,934	15.61%
Portuguese 2012	691,370	71,186	68,165	*17* :9,295	*18* :7,758	16.54%

Columns from left to right are: **Networks** = nontrivial category co-occurrence networks without isolated categories and feeble category edges; **#Category Edges** = total number of category edges ; **#Categories** = total number of categories; **Size C1** = number of categories in the largest cluster at threshold 2; ***t*1:Size C1'** = the first weight threshold *t* value before the largest cluster split (e.g. "426:4,276" = there are 4,276 categories in the largest cluster at threshold *t=426*); ***t*2:Size C1** = the second weight threshold value *t*2 = *t*+1 where the largest cluster is separated into smaller clusters (e.g. "427:2,675" = there are 2,675 categories in the largest cluster at threshold *t=427*); **%Size C1-Dropped** = percentage of the first largest cluster size has dropped significantly at the separating points ($t1 \leq t \leq t2$). Note that the network properties are the representative figures for the weighted category where $t \geq 2$.

Table 5.3 Comparison of the dropping sizes among the largest clusters for multiple category co-occurrence networks

While, for the 2010 edition in chart (a), after the giant cluster split, there are around 60 and 40 percent of categories for the two largest clusters. When checking those graph properties presented in Table 5.1, the volumes of pages and categories have remained constant. It is noticeable that the critical threshold values at which the giant splits for each of the three English networks increases from `427` to `2695` and `3046`; In contrast, the percentage of the giant clusters' size decreases. Possibly, when removing a few edges of the larger network with a higher threshold (lots of pages), the connected categories in the largest cluster are disconnected. However, the giant splitting size in the larger network is not as obvious as the smaller network, e.g. comparing the 2012 edition to the smaller scale of 2010 edition. For example, the 2011 and 2012 networks have the critical threshold t about 7 times of the 2010 edition, but the percentage of the giant split is lower. An explanation would be the disconnected categories are fragments (as tiny clusters) and many are isolates. This can be checked using Table 5.1. Similarly, the English 2015 has the significant point of the largest cluster separation as shown in Figure 5.11.

The giant splitting for the English 2015 graph as shown in Figure 5.11, is now represented in a better visualization, see Figure 5.12 using Pajek [197]; The largest cluster has separated into two large subclusters, but the rest of the small clusters are left out. Figure A.4 in Appendix A is a representation of the graph's global view for the threshold 3776. While, a close up view for the largest clusters is presented in the below pictures visualize for the threshold $t = 3777$. In addition, the English graph editions 2010-2012 before the giants split are represented in Figures A.1, A.2 and A.3, respectively.

Furthermore, not only is the largest cluster separating behaviour exhibited for the English editions, but it was also found in other languages such as French, Russian, German, Italian, and Portuguese. The log-log plots of the largest cluster for different weight threshold values, presented in a single chart in Figure 5.13 is shown for those five language editions of co-occurrence networks. The French edition has the significant drop in the size of the largest

cluster and higher threshold of the separating point compared to the others. A consequence of the significant separating points of the largest cluster for each network can be checked with Table 5.3; The table presents the number of categories, the largest cluster sizes at threshold 2 and range of t to t+1 and the percentage of dropped sizes of the largest cluster. Compared to the other non-English editions the French network has the highest figures of 20% size dropped when the threshold t increased from 150 to 151, while, the separation has appeared at the low threshold values $16 \leq t \leq 18$ and has similar percentages of the dropping sizes in the Italian and the Portuguese networks. The German is similar to the Russian network, and has more than one separation.

The giant splitting phenomenon in English network 2015 as shown in Figure 5.11 has appeared significantly, where the largest cluster with 1,383 categories at threshold 3776 has divided into smaller category clusters of size 1,051 and 327 at threshold 3777. The network is taken for further analysis, even though it is much larger, it has a smaller ratio of the separation than the others versions examined, i.e. only about 24% (2010 edition dropped almost 40%). This means its largest cluster has the most categories connected together. Also, using this latest version, the result can be interpreted with the current English Wikipedia. There are two goals to understand this behaviour: (I) what caused the giant splitting and (II) what the insights of the three largest clusters are after the split. The next subsections will approach these.

(I) What caused the giant splitting

The analysis in this stage is interested in finding what caused the separation of the largest clusters. Recalling from the first chapter, the Wikipedia categories are distinguished into two types; The 'administrative' (non-articles) categories, in short 'admin-categories' include the maintenance category pages indicating the article's status for the maintenance purposes and the 'Content' (articles) categories. The fact is that recently Wikipedia has been maintaining

its quality, and contributing many more projects to improve this, such as assessing quality, grade and importance of article-pages. This is the reason why it has a large growth of categories, fundamentally, they are the administrative types which also connect to the content categories. This assumes that most categories are connected together by a large bundle of admin-categories. Increasing a very high number of (admin-category) pages such as the maintenance categories can cause the large cluster to split. To test the assumption, an experiment was conducted as a proof of concept on the co-occurrence network. By stripping the maintenance categories from the whole original category network, then a pure content network can be obtained and the clustering results in relationship of administrative and content categories can be compared. The two versions of the category networks are constructed.

1. ***Version1*: content-administrative** category network is where the admin-categories were eliminated from the three observed largest category clusters at the critical weight threshold between 3776 and 3777.

2. ***Version2*: content** category network is where the admin-categories were eliminated entirely from the original (nontrivial) category network.

To obtain those two networks, first, a maintenance category rule-base must be created by using a few indicating keywords of the administrative categories used in [29, 32, 276] to purify the content categories. Also, a few other keywords such as 'maintenance', 'Wikipedia-backlog', 'Hidden-categories', 'Tracking-categories', 'Container-categories', and 'Very-large-categories' were added in. Next, the Wikipedia category's URLs were scanned for a corresponding category page by using each category title in the network toward the end of the URL-head 'category:'. Then the category page in HTML format was scanned for the keywords, and it was classified into ***content*** or ***admin-category***. The categories that contain a few keywords indicated as 'assessment category' (assessing quality, grade and importance of articles) such as '-ListClass-' and '-importance-', were also added into the rule-base.

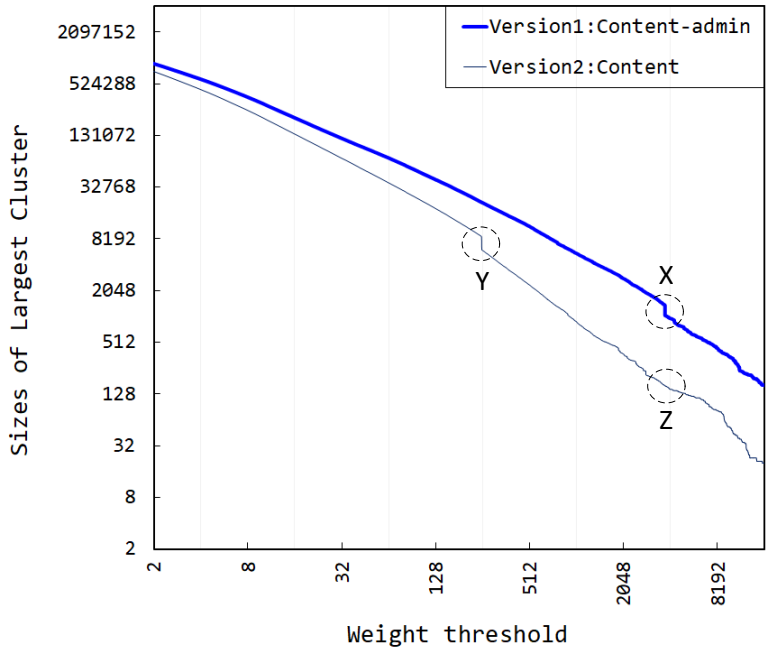


Figure 5.14 Log-log plots-comparison the largest cluster sizes for different weight threshold values of the two versions for the English Wikipedia category co-occurrence networks 2015

The analysis focused on finding whether the clusters separation depend on the maintenance category connectivity. Figure 5.14 shows the three focus points:

X = content category cluster of the network *version1* ($t1$=3776, $t2$=3777)

Y = content category cluster of the network *version2* ($t1$=252, $t2$=253)

Z = content categories of the network *version2* ($t1$=3776, $t2$=3777)

The chart of the log-log plots in Figure 5.14 shows the relationship of the largest cluster sizes versus the weight threshold for the two versions. *Version1*'s plots for the distribution of the mix types of categories are higher than *version2*, the pure content categories.An interesting point is that *version2*, content categories are still divided into subclusters even though the admin-categories were already removed. However, the significant splitting point of the giant cluster exhibits at the lower weight threshold (i.e. between the threshold 252 and 253) than *version1*'s. The two largest cluster separation is somewhat related to the connectivity among content categories; One shall reveal the insights of those category clusters.

Category Networks	Splitting Points	Threshold Values	C1		C2		C3	
			Page types	***n***	**Page types**	***n***	**Page types**	***n***
Version1: Content-admin Removed admin-categories at split point	X	$t1$=3776	Text-pages Talk-pages	90 66	Image-pages Audio-pages	26 1	Talk-pages about Food and drink	7
		$t2$=3777	Text-pages	85	Talk-pages	62	Image-pages Audio-pages	26 1
Version2: Content Removed admin-categories entirely	Y	$t1$=252	Text-pages Talk-pages	5526 382	Image-pages Audio-pages	184 1	Text-pages about Treaties of countries	30
		$t2$=253	Text-pages	5383	Talk-pages	382	Image-pages Audio-pages	184 1
	Z	$t1$=3776 $t2$=3777	Text-pages	102	Image-pages Audio-pages	29 1	Talk-pages	8

Columns from left to right are: **Category Networks** = obtained English category co-occurrence networks 2015 for two versions to be compared; **Separating Points** = area of cluster split at specific threshold values (e.g. 'X'=cluster split in the threshold range 3776 to 3777); **Threshold Values** = weight threshold values between $t1$ = t and $t2$ = t+1 where the large cluster split; C1 = the largest cluster; C2 = the second-largest cluster; C3 = the third-largest cluster; For each of the three largest clusters, a cluster contains categories that **Page Types** = types of the article pages such as text-pages, talk-pages, image-pages and audio-pages and the ***n*** = total number of categories.

Table 5.4 Comparison of category members between the two versions of category networks at where the category clusters separate for English category co-occurrence networks 2015

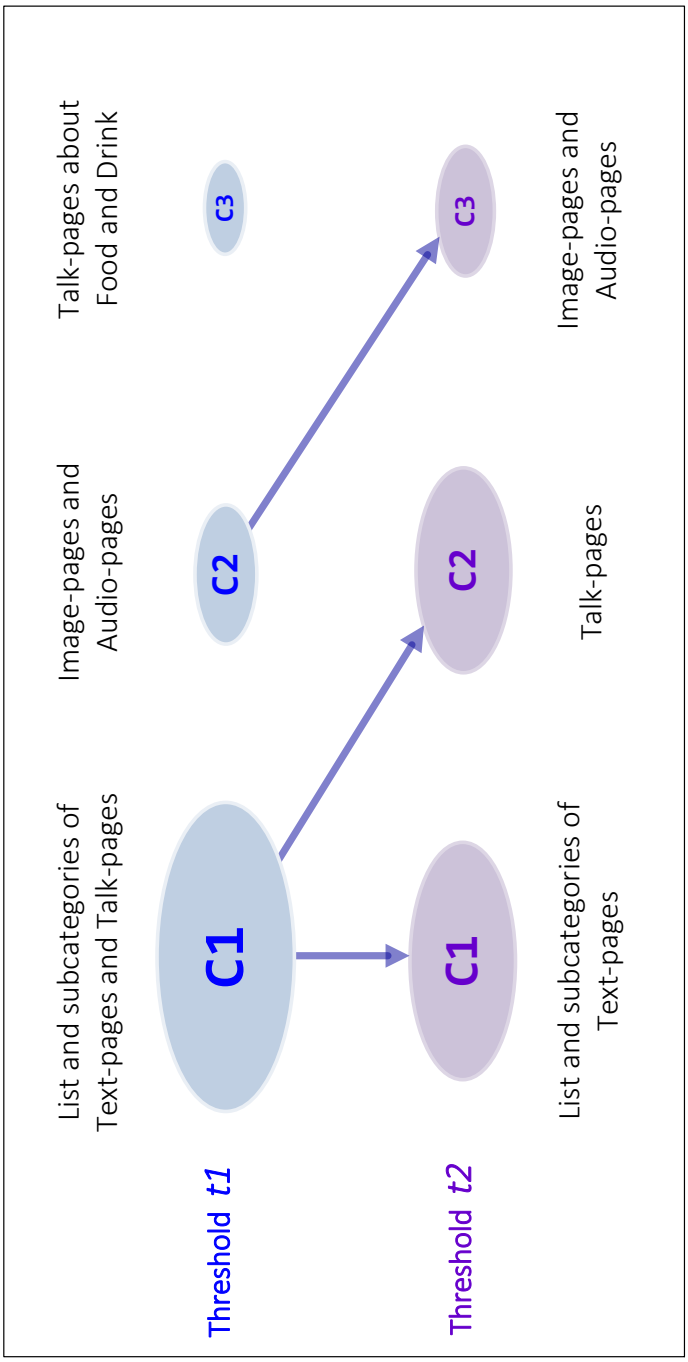


Figure 5.15 Distinction of category page types of the three largest clusters at the cluster separation between the weight threshold $t1$ and $t2$ for the English Wikipedia category co-occurrence network 2015

Categories: lists of text-pages and subcategories of text-pages					
No.	**Text-pages Lists**	**No.**	**Text-pages Lists**	**No.**	**Subcategories of text-pages**
1	1934 births	30	1980 births	1	American comedy films
2	1936 births	31	1981 births	2	American drama films
3	1937 births	32	1982 births	3	American film actresses
4	1939 births	33	1984 births	4	American male film actors
5	1940 births	34	1986 births	5	American male television actors
6	1941 births	35	1987 births	6	American television actresses
7	1942 births	36	1988 births	7	Association football defenders
8	1945 births	37	1990 births	8	Association football forwards
9	1946 births	38	1991 births	9	Association football goalkeepers
10	1947 births	39	1992 births	10	Association football midfielders
11	1949 births	40	1993 births	11	Black-and-white films
12	1950 births	41	1995 births	12	Brazilian footballers
13	1954 births	42	20th-century American actresses	13	British films
14	1957 births	43	20th-century American male actors	14	English footballers
15	1958 births	44	20th-century American musicians	15	English-language albums
16	1959 births	45	20th-century American novelists	16	English-language films
17	1962 births	46	20th-century women writers	17	German male writers
18	1963 births	47	Broadcast call sign disambiguation pages	18	Harvard University alumni
19	1964 births	48	Bundesliga players	19	Hindi-language films
20	1965 births	49	COIBot LinkReports for redirect sites	20	Human proteins
21	1969 births	50	Guggenheim Fellows	21	Indian films
22	1970 births	51	Human name disambiguation pages	22	Insects of Europe
23	1971 births	52	IUCN Red List data deficient species	23	Italian films
24	1973 births	53	IUCN Red List least concern species	24	Local COIBot Reports
25	1974 births	54	Letter-number combination disambiguation pages	25	Major League Baseball pitchers
26	1975 births	55	Stale Local COIBot Reports	26	Moths of Africa
27	1976 births	56	Villages in Turkey	27	Moths of Europe
28	1978 births	57	Wikipedia sockpuppeteers	28	Place name disambiguation pages
29	1979 births			29	Rivers of Romania
				30	Russian footballers
				31	The Football League players
				32	Villages in the Czech Republic
				33	Vulnerable plants

Table 5.5 *Version1-pointX*−category members of the largest cluster (C1) at threshold 3776

Categories: talk-pages e.g. task force/work group articles, article without infobox, users-bot report and other talk-pages			
No.	**Talk-pages: task force articles**	**No.**	**Talk-pages: work group articles**
1	Aerospace biography task force articles	34	Actors and filmmakers work group articles
2	British cinema task force articles	35	American animation work group articles
3	Documentary films task force articles	36	Animated films work group articles
4	English non-league football task force articles	37	Animated television work group articles
5	Episode coverage task force articles	38	Arts and entertainment work group articles without infoboxes
6	Film awards task force articles	39	Comics creators work group articles
7	Football in Brazil task force articles	40	DC Comics work group articles
8	Football in England task force articles	41	Marvel Comics work group articles
9	Football in France task force articles	42	Military biography work group articles
10	Football in Germany task force articles	43	Musicians work group articles
11	Football in Italy task force articles	44	Peerage and Baronetage work group articles
12	Football in Scotland task force articles	45	Politics and government work group articles needing infoboxes
13	Football in Spain task force articles	46	Royalty work group articles
14	Football in Sweden task force articles	47	Science and academia work group articles
15	Geography of Brazil task force articles	48	Sports and games work group articles
16	German military history task force articles	49	United States comics work group articles
17	History of Russia task force articles	**No.**	**Talk-pages: Articles without infobox**
18	Human geography of Russia task force articles	50	India articles without infoboxes
19	Italian cinema task force articles	51	School articles without infoboxes
20	Japanese military history task force articles	**No.**	**Talk-pages: users-bot report**
21	Maritime warfare task force articles	52	*COIBot LinkReports for redirect sites*
22	Paralympics task force articles	53	*Local COIBot Reports*
23	Political parties task force articles	54	*Stale Local COIBot Reports*
24	Politics and law of Russia task force articles	55	*Wikipedia sockpuppeteers*
25	Russian, Soviet and CIS military history task force articles	**No.**	**Other talk-pages**
26	Silent films task force articles	56	Accepted AfC submissions
27	Soccer in the United States and Canada task force articles	57	American cinema articles needing an image
28	Sports and games in Russia task force articles	58	Biography articles with comments
29	Sports in Brazil task force articles	59	College baseball articles
30	Technology and engineering in Russia task force articles	60	Old-time Base Ball articles
31	United States military history task force articles	61	Pages translated from German Wikipedia
32	Women's football task force articles	62	Portal-Class United States articles of NA-importance
33	World War I task force articles	63	Wikipedia requested photographs of actors and filmmakers
		64	Wikipedia requested photographs of military-people
		65	Wikipedia requested photographs of people
		66	Wikipedia requested photographs of scientists and academics

Table 5.6 (Continued) *Version1-pointX*−category members of the largest cluster (`C1`) at threshold 3776

No.	Category: image-pages
1	Album covers
2	Author died more than 100 years ago public domain files
3	Book covers
4	Cc-by-sa-3.0,2.5,2.0,1.0 files
5	CC-zero files
6	Copy to Wikimedia Commons reviewed by Sfan00 IMG
7	Copyright holder released public domain files
8	Creative Commons Attribution 2.5 files
9	Creative Commons Attribution 3.0 files
10	Creative Commons Attribution-ShareAlike 2.5 files
11	Creative Commons Attribution-ShareAlike 3.0 files
12	Fair use images of movie posters
13	Fair use magazine covers
14	Images in the public domain in the United States
15	Images published abroad that are in the public domain in the United States
16	Non-free comic images
17	Non-free posters
18	Other images that should be in SVG format
19	Public domain art
20	Publicity photographs
21	Publicity photographs with no terms
22	Radio station logos
23	Screenshots of films
24	Wikipedia license migration completed
25	Wikipedia license migration redundant
26	Wikipedia non-free historic files
No.	**Category: audio-page**
1	Wikipedia non-free audio samples

Table 5.7 *Version1-pointX*−category members of the second-largest cluster (C2) at threshold 3776 and the third-largest cluster (C3) at threshold 3777

No.	Category: talk-pages
1	Food and drink articles with incomplete B-Class checklists
2	Food and drink articles needing attention to referencing and citation
3	Food and drink articles needing attention to coverage and accuracy
4	Food and drink articles needing attention to structure
5	Food and drink articles needing attention to grammar
6	Food and drink articles needing attention to supporting materials
7	Food and drink articles needing attention to accessibility

Table 5.8 *Version1-pointX*−category members of the third-largest cluster (C3) at threshold 3776

Categories: lists of text-pages and subcategories of text-pages			
No.	**Text-page lists**	**No.**	**Text-page lists**
1	1934 births	42	20th-century American novelists
2	1936 births	43	Broadcast call sign disambiguation pages
3	1937 births	44	Bundesliga players
4	1939 births	45	Guggenheim Fellows
5	1940 births	46	Human name disambiguation pages
6	1941 births	47	IUCN Red List data deficient species
7	1942 births	48	IUCN Red List least concern species
8	1945 births	49	Letter-number combination disambiguation pages
9	1946 births	50	Villages in Turkey
10	1947 births	**No.**	**Subcategories of text-pages**
11	1949 births	51	20th-century American actresses
12	1950 births	52	20th-century American male actors
13	1954 births	53	20th-century American musicians
14	1957 births	54	20th-century women writers
15	1958 births	55	American comedy films
16	1959 births	56	American drama films
17	1962 births	57	American film actresses
18	1963 births	58	American male film actors
19	1964 births	59	American male television actors
20	1965 births	60	American television actresses
21	1969 births	61	Association football defenders
22	1970 births	62	Association football forwards
23	1971 births	63	Association football goalkeepers
24	1973 births	64	Association football midfielders
25	1974 births	65	Black-and-white films
26	1975 births	66	Brazilian footballers
27	1976 births	67	British films
28	1978 births	68	English footballers
29	1979 births	69	English-language albums
30	1980 births	70	English-language films
31	1981 births	71	German male writers
32	1982 births	72	Harvard University alumni
33	1984 births	73	Hindi-language films
34	1986 births	74	Human proteins
35	1987 births	75	Indian films
36	1988 births	76	Insects of Europe
37	1990 births	77	Italian films
38	1991 births	78	Major League Baseball pitchers
39	1992 births	79	Moths of Africa
40	1993 births	80	Moths of Europe
41	1995 births	81	Rivers of Romania
		82	Russian footballers
		83	The Football League players
		84	Villages in the Czech Republic
		85	Vulnerable plants

Table 5.9 *Version1-pointX*−category members of the largest cluster (`C1`) at threshold 3777

Categories: talk-pages e.g. *task force/work group articles, article without infobox and other talk-pages*			
No.	**Talk-pages: task force articles**	**No.**	**Talk-pages: work group articles**
1	Aerospace biography task force articles	34	Actors and filmmakers work group articles
2	British cinema task force articles	35	American animation work group articles
3	Documentary films task force articles	36	Animated films work group articles
4	English non-league football task force articles	37	Animated television work group articles
5	Episode coverage task force articles	38	Arts and entertainment work group articles without infoboxes
6	Film awards task force articles	39	Comics creators work group articles
7	Football in Brazil task force articles	40	DC Comics work group articles
8	Football in England task force articles	41	Marvel Comics work group articles
9	Football in France task force articles	42	Military biography work group articles
10	Football in Germany task force articles	43	Musicians work group articles
11	Football in Italy task force articles	44	Peerage and Baronetage work group articles
12	Football in Scotland task force articles	45	Politics and government work group articles needing infoboxes
13	Football in Spain task force articles	46	Royalty work group articles
14	Football in Sweden task force articles	47	Science and academia work group articles
15	Geography of Brazil task force articles	48	Sports and games work group articles
16	German military history task force articles	49	United States comics work group articles
17	History of Russia task force articles	**No.**	**Talk-pages: Articles without infobox**
18	Human geography of Russia task force articles	50	India articles without infoboxes
19	Italian cinema task force articles	51	School articles without infoboxes
20	Japanese military history task force articles	**No.**	**Other talk-pages**
21	Maritime warfare task force articles	52	Accepted AfC submissions
22	Paralympics task force articles	53	American cinema articles needing an image
23	Political parties task force articles	54	Biography articles with comments
24	Politics and law of Russia task force articles	55	College baseball articles
25	Russian, Soviet and CIS military history task force articles	56	Old-time Base Ball articles
26	Silent films task force articles	57	Pages translated from German Wikipedia
27	Soccer in the United States and Canada task force articles	58	Portal-Class United States articles of NA-importance
28	Sports and games in Russia task force articles	59	Wikipedia requested photographs of actors and filmmakers
29	Sports in Brazil task force articles	60	Wikipedia requested photographs of military-people
30	Technology and engineering in Russia task force articles	61	Wikipedia requested photographs of people
31	United States military history task force articles	62	Wikipedia requested photographs of scientists and academics
32	Women's football task force articles		
33	World War I task force articles		

Table 5.10 *Version1-pointX*−category members of the second-largest cluster (C2) at threshold 3777

Categories: list of text-pages					
No.	**Text-page lists**	**No.**	**Text-page lists**	**No.**	**Subcategories of text-pages**
1	1934 births	32	1982 births	71	20th-century American actresses
2	1936 births	33	1984 births	72	20th-century American male actors
3	1937 births	34	1986 births	73	American comedy films
4	1939 births	35	1987 births	74	American drama films
5	1940 births	36	1988 births	75	American film actresses
6	1941 births	37	1990 births	76	American male film actors
7	1942 births	38	1991 births	77	American male television actors
8	1945 births	39	1992 births	78	American television actresses
9	1946 births	40	1993 births	79	Association football defenders
10	1947 births	41	1995 births	80	Association football forwards
11	1949 births	42	***1935 births***	81	Association football goalkeepers
12	1950 births	43	***1938 births***	82	Association football midfielders
13	1954 births	44	***1943 births***	83	Black-and-white films
14	1957 births	45	***1944 births***	84	Brazilian footballers
15	1958 births	46	***1948 births***	85	British films
16	1959 births	47	***1951 births***	86	English footballers
17	1962 births	48	***1952 births***	87	English-language films
18	1963 births	49	***1953 births***	88	Human proteins
19	1964 births	50	***1955 births***	89	Indian films
20	1965 births	51	***1956 births***	90	Insects of Europe
21	1969 births	52	***1960 births***	91	Italian films
22	1970 births	53	***1961 births***	92	Moths of Africa
23	1971 births	54	***1966 births***	93	Moths of Europe
24	1973 births	55	***1967 births***	94	Place name disambiguation pages
25	1974 births	56	***1968 births***	95	Rivers of Romania
26	1975 births	57	***1972 births***	96	Russian footballers
27	1976 births	58	***1977 births***	97	The Football League players
28	1978 births	59	***1983 births***	98	Villages in Turkey
29	1979 births	60	***1985 births***	99	Vulnerable plants
30	1980 births	61	***1989 births***	100	***American films***
31	1981 births	62	***1994 births***	101	***Articles with hCards***
				102	***Articles created via the Article Wizard***
No.	**Category: other text-page lists**				
63	Broadcast call sign disambiguation pages				
64	Bundesliga players				
65	IUCN Red List data deficient species				
66	IUCN Red List least concern species				
67	Letter-number combination disambiguation pages				
68	***Human name disambiguation pages***				
69	***IUCN Red List endangered species***				
70	***IUCN Red List vulnerable species***				

Table 5.11 *Version2-pointZ*−category members of the largest cluster (`C1`) at the threshold range of 3776 to 3777

No.	Category: image-pages
1	Album covers
2	Author died more than 100 years ago public domain files
3	Book covers
4	Cc-by-sa-3.0,2.5,2.0,1.0 files
5	CC-zero files
6	Copy to Wikimedia Commons reviewed by Sfan00 IMG
7	Copyright holder released public domain files
8	Creative Commons Attribution 2.5 files
9	Creative Commons Attribution 3.0 files
10	Creative Commons Attribution-ShareAlike 2.5 files
11	Creative Commons Attribution-ShareAlike 3.0 files
12	Fair use images of movie posters
13	Fair use magazine covers
14	Images in the public domain in the United States
15	Images published abroad that are in the public domain in the United States
16	Non-free comic images
17	Non-free posters
18	Other images that should be in SVG format
19	Public domain art
20	Publicity photographs
21	Publicity photographs with no terms
22	Radio station logos
23	Screenshots of films
24	Wikipedia license migration completed
25	Wikipedia license migration redundant
26	Wikipedia non-free historic files
27	***Image of video covers***
28	***Non free logos***
29	***Screenshots of television***
No.	**Category: audio-page**
1	Wikipedia non-free audio samples

Table 5.12 *Version2-pointZ*−category members of the `C2` at *t* 3776-3777

No.	Category: talk-pages
1	Automatically assessed Football articles
2	Politics and government work group articles needing infoboxes
3	Start-Class National Register of Historic Places articles of Low-importance
4	Wikipedia requested photographs of politicians and government-people
5	College baseball articles
6	Football in England task force articles
7	Old-time Base Ball articles
8	Paralympics task force articles

Table 5.13 *Version2-pointZ*−category members of the `C3` at *t* 3776-3777

(II) Insights of Category Clusters

At this stage, the analysis is to reveal the insights of the three largest clusters after the split. To understand more about the giant splitting, an investigation is conducted to find what is the difference of the content-categories within each of the three clusters at the splitting points? Each of the three largest clusters for both versions in particular the points `X` and `Y`, and also compares point `Z` with point `X` has been focused. The differences among those categories within the clusters have revealed: (1) What are the categories within the three largest clusters at points `X` and `Y`? (2) Is the cluster separation behaviour of the two versions at point `X` and `Y` different? and (3) Do the content-categories in point `Z` belong to the cluster at point `X`?

A summary of the findings is represented in Table 5.4. An interpretation for the summary table is represented in Figure 5.15 for $t1$ and $t2$ ($t1$+1) in the following Tables and Figure 5.12 to minimise confusion. The three findings regarding the giants splitting for the two network versions are revealed as follows.

(1) Distinctive categories within each of the three clusters

This part reveals "*what the categories within the three largest clusters at points X and Y are*". An interesting finding is the distinctive categories within each of the three clusters for the two versions depends on the types of category pages, e.g. text, talk, image and audio. A comparison of the different category (category-pages) types in each cluster is provided in Table 5.4 for those splitting points in Figure 5.14. For example, considering *version1* at point X, these observed category-pages can be categorised into several types either 'text-pages', 'talk-pages', 'image-pages' or 'audio-pages'.

Considering *version1*, it can be seen that each of the largest clusters between thresholds 3776 and 3777 (`C1`-$t1$-3776 and `C1`-$t2$-3777) contains the 'list' (list of category pages) of text-pages (category-articles) and also 'subcategories'. The 'List of text-pages' cover

a range of subjects about people who were born in a certain year, places and events and subcategories related to the text-pages. The detail of the categories in the largest cluster (C1) in the ranges of those thresholds are presented in Table 5.5 continuing to Table 5.6. It is significant that the largest cluster (C1-*t*1-3776) contains 'text-pages' and 'talk-pages' categories. Whilst, the category members in the second-largest cluster (C2) are 'image-pages' and an 'audio-page'as presented in Tables 5.7 and 5.8. The categories in the first largest subcluster C1-*t*2-3777 as shown in Table 5.9 were all the members of the largest cluster C1-*t*1-3776 as shown in Tables 5.5 and 5.6. The articles in a specific subject can be found in the third-largest cluster at the threshold 3776. For instance, 'talk-pages' categories in this cluster are about 'food and drink' supported by the 'Wikiproject Food and Drink'. Table 5.8 displays these categories.

In the same way, the different kinds of category pages as explained previously for *version1* can also describe the distinctive categories of the clusters for *version2* as summarised in Table 5.4. Unfortunately, the cluster sizes for *version2* (where the giant was split between 252 and 253) for the three largest clusters are large. Their sizes are about thousands, and all category titles cannot be displayed in a table as presented for *version1*.

(2) Comparison of the separating clusters' pattern

This part reveals whether "the cluster separation behaviour of the two versions at point X and Y is different". The three largest clusters of the two versions are observed. The insights revealed that there are three similar regularities of the clusters' separation.

First, two subclusters derived from the largest cluster

It was observed that the three largest clusters are the two subclusters C1 and C2 at threshold *t*+1, and are completely derived from the largest cluster C1 at threshold *t*. The proportion of the derivation from the largest cluster are about 60% for the largest subcluster and 40% for

the second-large subcluster (at threshold 3777), in `Version1`. Whilst, the ratio of *Version2* is `90%` and `10%` for the two subclusters, respectively (at threshold 523). Table 5.4 provides the information where the percentages can be verified. The types of these categories are talk-pages which are generally either 'task force articles' or 'work group articles', and the rest are other talk pages. Note that a task force is generally set up on a subpage of the parent project page, and an infobox is a template used to collect and present a subset of information about its subject.

Second, the difference between the two subclusters

It is observed that the two versions have the similar behaviour of deriving the subclusters from the largest cluster and forming the category members by certain kinds of categories. It can be seen that the two largest subclusters contain different categories. The text-pages in the largest subcluster and all talk-pages in the second-large subcluster belong to the largest cluster. Whilst, all the categories (image and audio pages) of the third cluster at the t+1 came from the second-largest cluster at t. This can be confirmed with Table 5.4 which presents the kinds of category-pages and the amount of the category members within each of the three observed clusters. To illustrate, for *version1-X* at the threshold 3777, two subclusters are divided from the largest cluster (`C1`) at the threshold 3776. The third cluster (`C3`) at threshold 3777 has become the second cluster at threshold 3776. The second subcluster contains mostly the list of talkpages supported by different Wikiprojects covering various subjects, such as military, films and so forth.

Third, deriving the third subcluster

The third-largest subcluster (`C3`) at threshold t+1 was gained from the second-large cluster (`C2`) at threshold t. Table 5.7 presents these two clusters. The cluster members are 'image-article pages' and an 'audio-page', observed to support the content 'text-pages'

articles in terms of completing the context article pages (with multimedia content). While the 'talk-pages' are observed to be more concerned with the contributions of the content's quality from the registered authors. However, the cluster sizes of the three largest clusters are large, and all category titles cannot be displayed in a table like it was for *version1*. In summary, it can be observed that *version1-X* and *version2-Y* exhibit the similar behaviour of the cluster separation, but the cluster sizes of *version2* are larger than *version1*.

(3) Similarity of *Version1-X* and *Version2-Z*

This part is to analyse the relationship of clusters' members between the separation at threshold 3776 *version1-X* and the non separation point from 3776 to 3777 in *version2-Z*. The content categories in each of the three clusters in the content *version-2* would probably belong to the clusters in the admin-content *version1*. The categories within the three clusters of the content category network are presented in Table 5.11, 5.12 and 5.13. It can be seen that the categories in the three clusters are the same types in both versions such as text-pages, image and audio pages and talk-pages. It was observed that most content categories in each of the two largest clusters belong to the clusters of the admin-content version. The category titles in italic-bold highlighted as the content categories that do not belong to the clusters in the admin-content version. There are 70% of the same categories within the largest cluster for both networks shown as 75 categories in Table 5.11. For the second-largest cluster, there are 90% where the clusters of the two versions share the same categories (see Table 5.12). Unlike, the category within the third cluster that does not belong in the clusters of the mixed categories network as shown in Table 5.13.

Furthermore, the categories in the largest cluster where the subcategories of the text-pages were revealed in Tables 5.5, 5.9 and 5.11 contain the lists of people by occupations such as actresses, actors and football players. The pages of these popular people in the encyclopedia are presented in their name lists. Regarding the awareness of the gender

bias mentioned in the introduction, when looking into the gender equality for popular film people, women are involved in films more than men. It can be seen that the categories *American film actresses* contain 200 pages and *American male film actors* contains 22 pages (last reviewed in July 2019). When considering the categories revealed in the three tables, there have been more articles on men (thousands) who are association football players than women (hundreds) for the content category the *association football players by position*[1]. For example, the content of male football players has presented in separate subcategories such as defenders, forwards, goalkeepers and midfielders, in total 16 categories, while for *women's association football players*[2] there are only 5 subcategories inside the category association football players by position. It can be explained that females have participated in association football less than males, despite the amount of wiki-pages and categories represented for those women and men.

5.5 Chapter Summary

The proposed analytic framework was used in clustering several category co-occurrence networks in Wikipedia. The novel findings in the co-occurrence graphs, the comparison results between m-core and k-core and the insights of the category hubs separation were presented.

First, the properties of the multiple category networks in Wikipedia were revealed. The results of the English and German editions were fairly similar; The growth of categories was significantly higher than the pages in both English and German networks. The behaviour of the new wiki pages which were contributed into the network tended to be categorised into related existing categories following the 'rich get richer' behaviour of a scale-free network.

[1] https://en.wikipedia.org/wiki/Category:Association_football_players_by_position (reviewed in July 2019)
[2] https://en.wikipedia.org/wiki/Category:Women%27s_association_football_players_by_position (reviewed in July 2019)

This was observed from the number of the isolated pages and categories and the growth of the category edges that have an exponential relationship to the weight threshold.

Second, using m-core clustering, all the observed properties regarding the cluster size exhibited the power-laws with respect to the weight threshold, and were unchanged over the period of time observed. As a result, there was a significant finding where the largest category cluster of each network was separated into subcategories when increasing the number of pages. This finding indicates the giant cluster to be split into a few category hubs for all language editions examined. However, this does not appear for the k-core.

Third, as mentioned above regarding the category cluster separation when the number of pages is increased, leads to the assumption that most categories would be connected by a large number of admin-category pages. It was tested by removing the admin-categories from the network, and performing the clustering. The results show that most categories are connected together by the admin-categories, and not only by their semantic relations.

Forth, the insights within the three category hubs are that the categories distinction depended on the types of category pages such as text, talk, image and audio, no matter if the categories were content or administrative categories. The largest sub cluster contained text article pages, and the second largest sub cluster contained only the talk-pages. While, the third sub categories were the image and audio article-pages.

Finally, the largest sub cluster contained a few lists of people by occupations such as the amount of text article pages of American actresses being more than male actors, while there are less pages about women's association football players than male players.

Chapter 6

Clustering Validations

Previously, the most significant finding was the appearance of the category hubs where the largest category cluster was divided into a few smaller clusters for each category graph. This chapter presents two validations for the clustering result. The first section is the validation of the category hubs result on the random permutations of the page-category graphs. The second section demonstrates how the categories within the hubs will be validated on the taxonomy graph. The final section is the summary of the chapter.

6.1 Validation on Random Graphs

The presence of the category hubs presented in Chapter 5, Section 5.2 suggested that the category graph's topology is scale-free, where the category connectivity was not growing randomly. An experiment in this section is to validate if the relationship of the categories is not formed in a random manner by testing it on a number of random page-category graphs. Before getting into the experimental content in detail, the definitions related to generating random graphs are given in the next page.

6.1.1 Definitions of the Random Graphs

Terminology that are relevant to the experiment on the random Wikipedia page-category graphs are described as follows.

Definition 10 `(Page-category edge list)`

Edge list sorted by page vertex of the original page-category graph following Definition 1 is the page-category edge list.

Definition 11 `(Random page-category edge list)`

A random page-category list is an edge list after permutating a category with a randomly selected entry to a new category in the page-category edge list, but the list of pages is fixed.

Definition 12 `(Multiple page-category edge)`

A page-category edge is a multiple edge if the same page vertex corresponds to the same category vertex.

Definition 13 `(Random graph)`

The random graph is a graph where all categories were permutated randomly and its random page-category edge list contains no multiple edge.

6.1.2 Generating the Random Graphs

The original English Wikipedia page-category graph 2010, where the edges are sorted by page labels and contains no multiple edges, is used for the graph permutation. Note that the isolated pages and categories (see Definition 2 and Definition 3) are eliminated from the original page-category graph as was explained in Chapter 4. Informally, a random graph

following Definition 13 is generated by permutating all categories from the page-category edge list (see Definition 10), but retaining all the pages; The category connections have been chosen at random. This means the connectivity of all categories does not depend on the links of the pages but they are randomly connected among themselves. This graph will then be analysed in the same way as the original graph using the t-component framework. Each random graph will be transformed into a category co-occurrence graph, and the categories will be clustered by the m-core clustering.

Algorithm 7: Wikipedia Page-category Graph Permutation

Input: a page-category graph
Output: a random graph *of* ***random page-category list*** without multiple edge

```
1 for each category in the page-category edge list do
2     Random page-category list ← permutated category from page-category edge list
3 repeat
4     for each unique page in the random page-category edge list do
5         if there is a multiple page-category edge then
6             Random page-category list ← permutated category from the page-category
                edge list
7 until the random page-category list contains no multiple edges;
8 return the random graph
```

Algorithm 7 presents the procedures to produce the random graph. The edge list of the input page-category graph is obtained by taking their list in original order. The first permutation graph' process shows at lines 1 to 2. It is permuting the category with a randomly selected entry in the category list (with replacement) of the page-category edge list, but the page list will remain. At this stage, the permutated categories list is replaced into the random page-category edge list, following Definition 11.

After the first permutation has been done, the obtained random graph must ensure that it contain no multiple edges as demonstrated in the procedure at lines 3 to 8 of

Algorithm 7. In order to remove any such duplicates another swap is performed, but only with the duplicates, and this is repeated until there are no multiple edges remaining. The graph permutation process is iterative and continues until all the categories have been swaped.

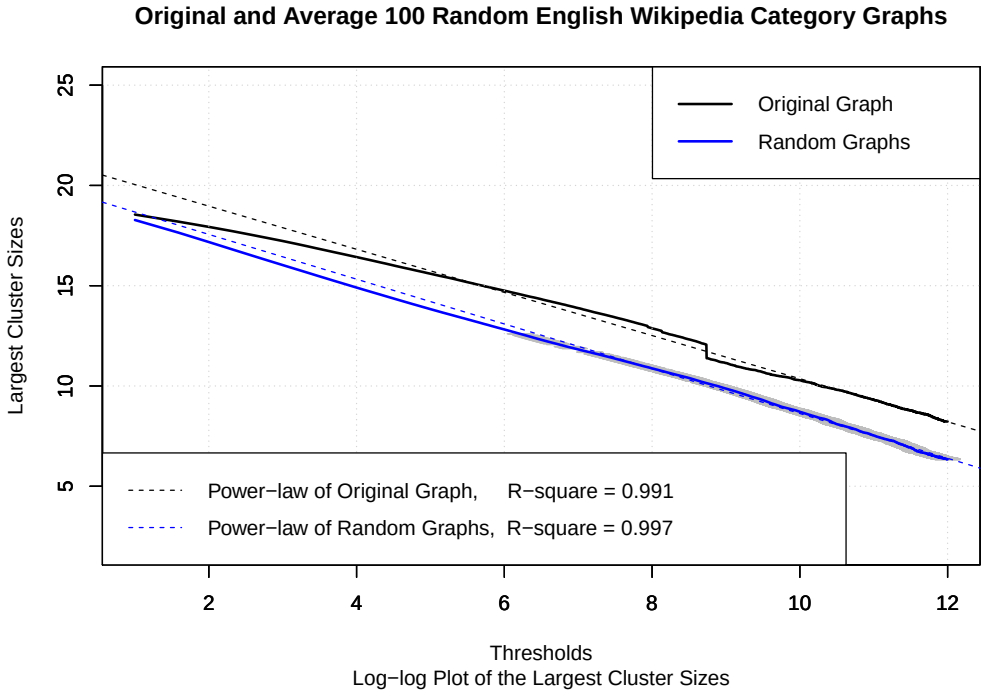


Figure 6.1 Log-log plots-comparison of original and average 100 random English Wikipedia category co-occurrence networks 2010 on number of categories of largest cluster

A hundred graph permutations were generated for the test. For the results of the first permutation, the average random graph contained less than 1% duplicates, approx three hundred thousand. For the next permutations, the number of the duplicates were less than 20,000, 2,000, and 10 duplicated categories. On average for each random graph, the program performed the permutation approximately six times before completing a final random graph containing no duplicates.

6.1.3 Validating the Cluster Result on the Random Graphs

The clustering result on the co-occurrence graph where the category connectivity was generated in a random manner is presented in Figure 6.1. The average sizes of the largest clusters for the hundred random graphs is plotted in log-log scale showing the error bars in the tiny shade, and are very small (see Figure 6.1). There is no difference between the random plots and the linear regression that is statistically fit well under the random graph test. Nevertheless, when comparing to the original graph, the random version has a much lower number for the largest cluster sizes. An explanation is that any category of the random graphs has similar possibility to be connected arbitrarily. Furthermore, it is significant that the random version does not show the category hubs separation phenomenon as the original graph. The random graph line is almost a straight line on the best fit for linear regression, and it has a steeper slope compared to the original version.

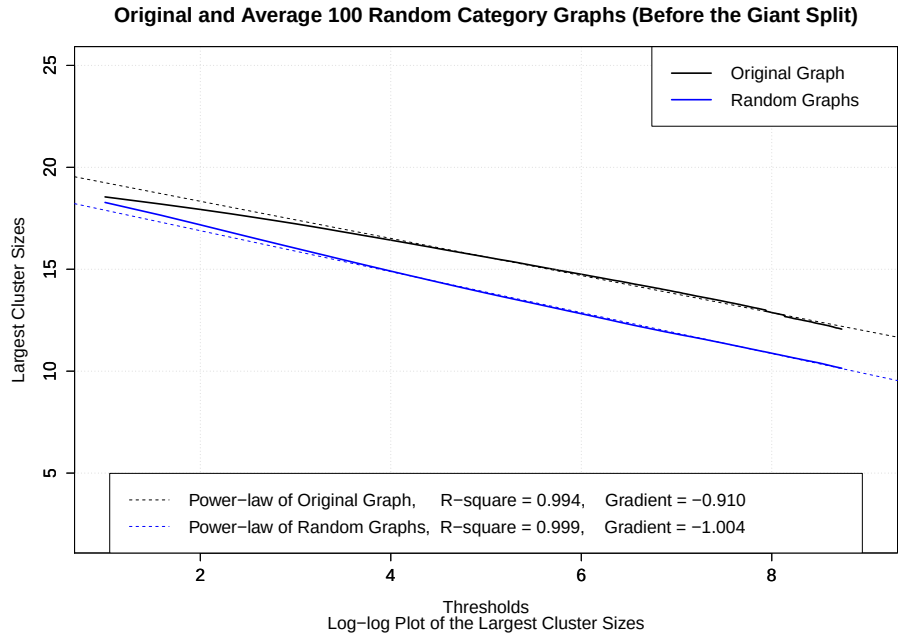


Figure 6.2 Log-log plots-comparison of two English Wikipedia category co-occurrence networks 2010: original and random versions (before the cluster separation)

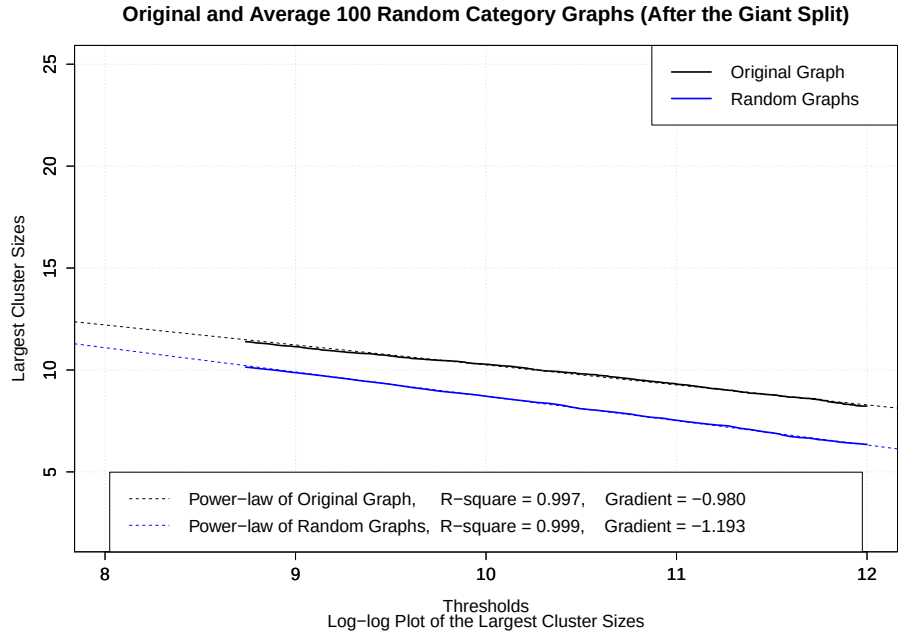


Figure 6.3 Log-log plots-comparison of two English Wikipedia category co-occurrence networks 2010: original and random versions (after the cluster separation)

From the original graph, the largest cluster sizes had dropped sharply as shown in Figure 6.1 and more detail in Figure 5.10 (a). The clustering result obtained from the original graph was validated on the average of the hundred random graphs. The validation is divided into two parts and custom confidence interval are established for the tests. The first part is before the largest cluster separation from the weight threshold 2 to 426, the results as shown in Figure 6.2. The second part is where the largest cluster divides into the smaller clusters from weight threshold 427 to 4096, and the result of this is shown in Figure 6.3. The two figures (Figure 6.2 and 6.3) are the comparison of the best fit in the linear regression lines for the original and the random versions, plotted on the log-log scale. The random graphs in both charts have a higher gradient compared to the original graph.

6.1.4 Conclusion

A two tailed test for threshold 2 to 246 and another one for threshold 247 to 4096 for the random graphs at confidence interval 1.25% and 98.75%, the gradient value are 1.11 and 1.12, respectively. For the original graph, the gradient value was 1.08 which is not fall in the rejection region; This means that the random graphs are not representative of the real English Wikipedia category graph. There was enough evidence that the categories in the original graph are not connected together randomly but by pages. The topology of the original Wikipedia category graph is not a random graph, but scale-free with a few category hubs, connecting most of the categories together. It can be concluded from the result of the experiment on the number of random graphs that the category hub phenomena found was real, and not artifact of the data.

6.2 Validation on a Taxonomy Graph

This section presents the validation of the co-occurrence graph clustering result where the category hubs were found on the taxonomy graph. The idea is to bring the category population from those hubs and test against a taxonomy benchmark graph which has not been used for the clustering. This experiment is to test consistency in the category structure of the co-occurrence with the taxonomy graphs by comparing distances within and between the category clusters. The distances are expected to follow the principle of clustering that there should be a high similarity within each cluster, and a high dissimilarity between the clusters.

Although, the co-occurrence and taxonomy graphs are the representation of the category structure originated from the English Wikipedia 2010, their constructive nature is different. Recalling that the co-occurrence graph is an authentic association in the graph database as

a ground truth network of all types of categories in a large collaborative thesaurus manner. Whilst, the taxonomy graph was built in the hierarchy tree manner which contains the hierarchical structure of content category connectivity. The fact is that the co-occurrence graph is considered as a simple graph, while the taxonomy graph is a multiple directed graph with loops. This certainly requires a graph modification to remove the unnecessary loops and multiples edges from the taxonomy graph before comparing with the co-occurrence graph. There are several procedures involving exploring and modifying the taxonomic graph. This modified taxonomy graph will be mapped with the co-occurrence graph. The few largest clusters are used for the graph structure comparison by comparing distances from the clusters to the taxonomy graph. To do this, the distances will be found by: 1) obtaining the intra distances within the two largest clusters, and the other sizes of clusters, i.e. the rest of categories and 2) obtaining the inter distances between each pairs of those clusters. The inter distances are expected to be larger than the intra distances.

The experiment focuses on the significant category hub separation of the Wikipedia co-occurrence graph 2010 where the three largest category clusters are taken as the category population for the comparison; The two largest category clusters of size `2,675` and `1,588` at the weight threshold 427 are separated from the largest cluster of size `4,276` at threshold 426. The original *Wiki-taxonomy graph*[1] (last reviewed in March 2019) as an external benchmark graph, a project of constructing Wikipedia Category Taxonomy 2010 [258]. It was clawed from the articles whose title began with 'Category:' in the wiki-text source and metadata embedded in XML format will be used for the validation. Before going straight into the validation procedures, there are necessary terms on the clustering results obtained from the co-occurrence graph and terminology relevant to the taxonomy graph used in the experiment that need to be described.

[1] http://wikicategory.sourceforge.net

6.2.1 Graph Terminology

Definition 14 `(Category arc)`

The category arc is a relation of a category pair with direction from where a parent category links to a child category within a directed graph.

Definition 15 `(Multiple category arc)`

A category arc may appear in a directed graph more than once. This same arc is called a multiple category arc.

Definition 16 `(Category loop)`

A category edge or a category arc is a loop if it links to itself.

Definition 17 `(Original taxonomy graph)`

The original taxonomy graph is a multiple directed category graph which has category vertices, multiple category arches, and allows category loops.

Definition 18 `(Category root)`

A category is a root if it is only a parent but not a child within the original taxonomy graph.

Definition 19 `(Category leaf)`

A category is a leaf if it has no children within the original taxonomy graph.

Definition 20 `(New taxonomy graph)`

The new taxonomy graph is a final graph modified from the original taxonomy graph where every category arc is converted to a category edge by removing the multiple arches and the loops.

Definition 21 `(Largest cluster)`

A category cluster is a largest cluster if it has the greatest size or highest number of connected categories from the others.

Definition 22 `(Second-largest cluster)`

The second-largest cluster is a category cluster that is smaller than the largest cluster but greater than the others.

Definition 23 `(Intra distance)`

The intra distance is a distance between a pair of categories within a cluster.

Definition 24 `(Inter distance)`

The inter distance is a distance between one to another category from different clusters.

Wikipedia taxonomic graph contains the information of graph traversal from a parent category to a child category. It is used to find the shortest path distance between the categories. The graph has the multiplicities and loops which are not necessary for that. There are two main procedures of the graphs data preparation: modifying and mapping the graphs as the following demonstrates.

6.2.2 Taxonomic Graph Modification

This stage presents a graph modification on the taxonomy graph which is initially a multiple digraph with loops (following Definition 17). This taxonomy graph will be converted to a simple graph in the same form as the co-occurrence graph (following Definition 4). To do this, first of all, the graph is explored and manipulated beforehand, so that it can be mapped to the co-occurrence graph.

In next page, Table 6.1 shows the taxonomy graphs in tables and their graph visualizations are represented in Figure 6.4 which the figure (a) represent for the table (a) and so on for (b). The original taxonomy graph shown in (a) containing two multiple arches `(1,3)` and `(1,5)` and a category loop `(8)`. This graph is converted into the new taxonomy graph in (b) without any multiplicities and loops. There were 741 multiple category arches and `1,174` loops in

the graph that were removed. At this stage, the new taxonomy graph is a simple digraph, and there are 960 category roots following Definition 18, and 287,136 category leaves following Definition 19.

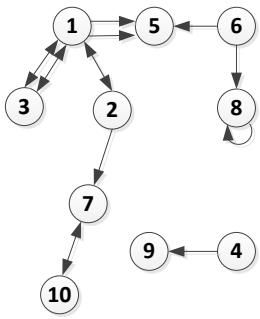
Parents	Children
1	2
1	3
1	***3***
1	5
1	***5***
2	1
2	7
3	1
3	***1***
4	9
6	5
6	8
7	10
8	8
10	7

(a) original taxonomy graph

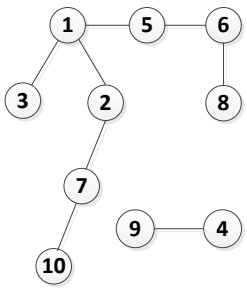
Category1	Category2
1	2
1	3
1	5
2	7
4	9
5	6
6	8
7	10

(b) new taxonomy graph

Table 6.1 Presentation of the taxonomy graphs:



(a) original graph



(b) new graph

Figure 6.4 Visualizations of the taxonomy graphs

6.2.3 Mapping Co-occurrence with Taxonomy Graphs

To be able to validate the clusters obtained from the co-occurrence graph with the taxonomy graph, the two category graphs are mapped into a lookup table to relabel the categories matched with the taxonomy graph. To this end, each of the categories was labeled with a category name, and it is used as the primary key to match a pair of different category IDs from both category graphs. The lookup mapped table from the two graphs is constructed as shown in Table 6.2.

MatchingIDs	**Taxonomy CategorylDs**	**Taxonomy Category Titles**
75624	1	Categories named after television series
125464	2	Futurama
406067	3	Works by Matt Groening
123183	4	Fox graph shows
85009	5	Comedy Central shows
193599	6	New York City in fiction
85110	7	Comic science fiction
18265	8	1930s
267041	9	World War II
.	.	.
.	.	.
529799	473630	Women in 17th-century warfare
-999999	473631	Chicken Plumage Patterns
-999999	473632	George Fulton
529804	473633	Jurisprudence academics
529490	473634	Zion, Illinois
247754	473635	Settlements established in 1901
529807	473636	Eating utensils
529813	473637	Scottish legal scholars
529808	473638	English legal scholars
529810	473639	Izuhakone Daiyuzan Line

Table 6.2 Lookup mapped table from the two graphs

Note that before performing the clustering, the category labels (numeric category IDs) in the co-occurrence graph were renamed and ordered to be operated easily. The category labels are therefore renamed back to the original ones. There are a few data preparation processes. The category titles's token formatting of both category graphs were not consistent such as

a few tokens by '_' and a few by ' -'. Instead, they were replaced with white space, ' ', and the missing values were replaced with 'N/A'.

The mapping process begins by searching through the category titles on both graphs to find any matching titles. There are 455,720 matched category titles from the two graphs, 80.24% in total. If there is a match in category IDs, 'Taxonomy CategorylDs' were added into the lookup table. Otherwise, '-999999' is placed as the unfound acknowledgment. The taxonomy and co-occurrence graphs contain 473,639 categories and 567,939 categories respectively. At this stage, the (matched) category lists following the Definition 21 and 22 are constructed as follows: the largest cluster contains 2,225 categories; the second-largest cluster has 1,060 categories and the other clusters contains 452,435 categories.

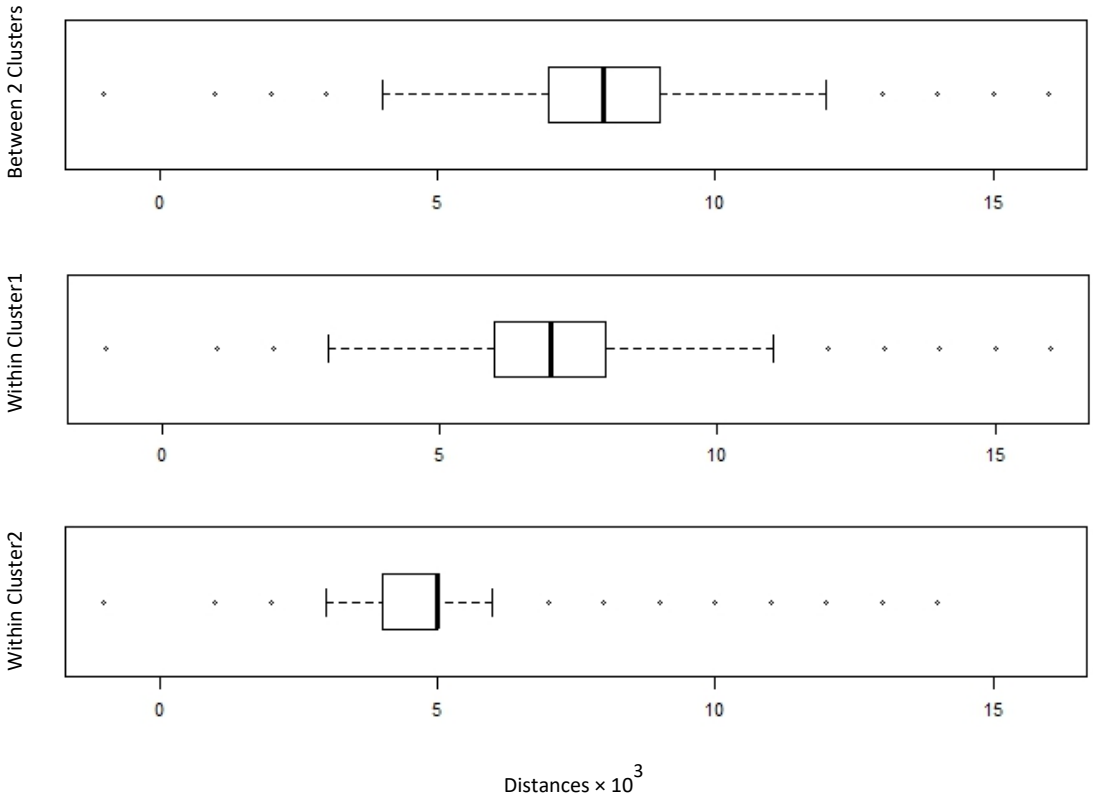


Figure 6.5 Distances of the two largest clusters

6.2.4 Validating the Cluster Results on the Taxonomy Graph

The two largest clusters within the t-filtered graph where t is 427 are used to validate the cluster results by comparing the shortest path distances between the categories. This is to find the intra distances within each of the two largest clusters and the inter distances between each pairs of clusters. The inter distances are expected to be larger than the intra distances, following the clustering principle. Note that the two largest clusters are more focused, and the other clusters with different sizes (mostly size 2) are considered as the other clusters. The approximate number of random categories is 200 from 2,225 for the largest cluster, 100 from 1,060 for the second-largest cluster, and 40,000 from 470,354 for the rest of the clusters. Here, the random category list for the two clusters are generated. They are randomised independently. Afterward, this generated random category list will be used to create the category pairs when measuring the similarities.

A shortest path distance is used as the similarity between a pair of categories. Algorithm 1 demonstrated in Chapter 2 is used to search the distance for a category pair as a single source search from one to all connected categories. The distances for each pair from different clusters, between the largest cluster and the second largest cluster are measured. The measurement is performed a possible unordered pair of categories (c_i, c_j) where c_i and c_j are from different clusters. The 2,358,500 inter distances are obtained between the two clusters where category c_i is from one cluster to c_j another cluster.

Distances	**Largest Cluster**	**Second-largest Cluster**
Largest cluster	5210.62	5334.17
Second-largest	-	3384.83

Table 6.3 distances matrix of the two largest clusters without sampling

The box plots as shown in Figure 6.5 is a representation of the distance matrix without sampling. The distances between the two clusters are large but it is not much larger compared to the intra distances within the largest cluster. The distances within each of the two clusters are measured for all pairs in each cluster. Within the largest cluster, there are totally 2,474,200 pairs with intra distances of 2,225 categories, while 561,270 distances are obtained from the 1,060 categories within the second-largest cluster. It can be seen in the distances shown in Table 6.3 that there is a tight membership within the second largest cluster with the intra distances 3,384.83 on average; This is far apart from another cluster with the average inter distance 5,334.17.

6.2.5 Conclusion

The main experimental work is to validate the clusters obtained from the co-occurrence with the taxonomy graph. The taxonomy graph was converted into the same type of co-occurrence graph. The test was done on the cluster splitting phenomenon where the largest category cluster was split into smaller clusters. For the validation clustering result tested with the taxonomy graph presented, it can be concluded that the two clusters and a groupof the other smaller clusters (considered as a cluster) were far apart from each other. The category pairs' distances within each of the two largest clusters were smaller, which also confirmed that the clusters were obtained correctly from the co-occurrence graph. All in all, the three clusters were far apart from each other. The relationship of categories in the second largest cluster is very close compared to the category relationship in the largest one and the other clusters. There was enough evidence to conclude that the clustering result where the two clusters are separated from the largest cluster follows the principle of clustering: The two clusters have small distances within them but the distances between the two are large. The outcome of the clusters validation confirmed that the obtained clusters from the co-occurrence graph are consistent to the taxonomy graph.

6.3 Chapter Summary

This chapter proved that the category co-occurrence structure is consistent to the taxonomy graph in the English version 2010 by validating the clusters of the co-occurrence with the taxonomy graph that was not used for the clustering. Before the validation, there were a few processes to prepare the taxonomy graphs for the test. As the formation of both graphs are different, the taxonomy graph was modified into a simple graph and the categories were mapped by their labels. The category members in each cluster were retrieved from the lookup matched table categories.

The validation was done by measuring all category pairs' distances within and between the clusters only where the category clusters split. There were only two sets of clusters used for this test, the two largest clusters. The taxonomy graph was used to validate the clusters, how far apart they would be, and how close are the category members within each of the two clusters.

In summary, the two clusters were far apart from each other, and the category pairs' distances within each of the two largest clusters were small. The test for the other clusters was done by sampling to find the distances within and from itself to the two largest clusters. The cluster validation results confirmed that the obtained clusters from the co-occurrence graph are consistent to the taxonomy graph by following the clustering principle.

Chapter 7

Summary and Future Work

This chapter provides the thesis summary of the proposed graph analysis methodology, the insights of the analyses and the validation of the novel findings. In addition, a few possible directions for the future research following the thesis are also addressed.

7.1 Summary of the Thesis

This thesis surveys research opportunities in Wikipedia mining for content-based, graph-based and hybrid-based analyses where the graph-based analysis is enabled to leverage text analysis and enrich semantic representation. The main focus of the study in this thesis is to analyse the Wikipedia category network which is represented as the co-occurrence graph, extrapolated from the pages and categories relationship. The investigation takes the whole connectivity of categories for granted as an empirical category graph analysis, but is not concerned in their textual relationship. This is to study its structure where all possible semantically relevant categories have been generated through the collaboration thesaurus mechanism of the Wikipedia categorisation system. The fact is that the size of the initial graph is large, and the overlapping relations among the pages and categories are complex,

this requires an appropriate analytic methodology. The challenges of this graph analysis are dealing with the scale and complexity of the graph due to the massive links of pages and categories. It motivates this thesis to research a graph analysis methodology to discover communities of Wikipedia's topics and analyse their relationship.

To give a concrete knowledge in building the thesis contributions on the analytical methodology, the background on Wikipedia, graph modelling, social network analysis and the cohesive subgroups is presented. It surveyed possible graph analysis methodologies and their justification for which approach would be appropriate to analyse the category networks. The comprehensive surveys on the cohesive subgroups, especially the k-core and m-core are provided. Furthermore, it includes a comprehensive literature survey on the Wikipedia graphs of what researchers have been studying so far in different applications and techniques. The most related work on the co-occurrence networks were reviewed and discussed.

Co-occurrence analysis is involved in diverse data sets where a co-occurrence graph with a single set of vertices (e.g. weighted graph) can be induced from a bipartite graph. The m-core which is concerned with co-occurrence quantification in a graph, such as edge frequencies in a weighted graph has been applied for various applications, known more as m-slice. However, to the best of my knowledge, the m-core has not been used to analyse the Wikipedia networks.

This thesis has introduced the t-component framework for the graph analysis, embedding the m-core as a component for constructing subgraphs by varying the shared pages number, threshold t of the category edges. Besides, a simple graph partitioning technique is used to handle the graph scale during the processes of transforming, filtering and clustering the categories on each subgraph in turn. In brief, a large page-category graph was divided into small subgraphs, and each subgraph was transformed into a co-occurrence subgraph as a weighted graph. Within each co-occurrence subgraph, the category edge cuts were

minimized and the category weighted-edges were filtered in order to obtain more dense graphs. The connected categories were identified, and finally any categories that can be connected from different subgraphs, were merged into a single graph.

The t-component was applied for the analyses in several editions of Wikipedia category co-occurrence graphs for different years and languages. The evolution of the graphs affirmed that the growth of the categories was higher than the pages over time observed. An assumption for this is there would be more admin-category pages created to maintain the quality of the articles in the encyclopedia. It has been found that the structure of the category graphs were stable; It has been seen that all distributions of the observed category cluster properties followed the power-laws having a declining exponential relationship with the weight threshold values. The most important finding was the distribution of the largest cluster's size exhibits the power-law with a slow fall, exponent averages approximately 1. It revealed that the largest cluster shrank significantly when the shared pages had risen to a critical threshold t, and a giant component was split into a few category hubs. This finding was validated using a permutation test where the categories were generated randomly but the pages were fixed. Despite the absence of the large cluster shrinking in the random graphs, it proved that the giant split phenomenon was real. The categories relationship in the co-occurrence graph is scale-free for having the power-law distribution with presence of a few category hubs. This can be diagnosed only by the m-core but not the k-core.

The dominant categories after the giant split were used to validate the clustering result on the taxonomy tree graph. The result showed that the category structure of the two graphs was consistent by comparing the distances within and between the category clusters. An analysis was carried out to reveal what caused the giant cluster to be split. The categories growth is higher than the pages since there are more admin-categories established to maintain the article pages. Removing a few category edges, therefore, causes a reduction in the size of the giant cluster, obviously separating into two cluster hubs.

7.2 Directions for Future Research

Future work would probably be comparing the co-occurrence graph with taxonomy graph where both graphs contain only content categories. If their structures are similar then the Wikipedia semantical topics gained from the co-occurrence graph would be useful for IR research field. In text classification, for example, the relevant subjects are preferably constructed in the taxonomy fashion [12, 54, 58], in fact the semantic graph is often used in text classification, e.g. [31]. This would be used for NLP, IR and network science research communities, and also the analysis approach may be helpful to relevant work on different domains.

Another direction for future study would be research in estimating how close the categories are together, which can define category similarity in many different ways. The cosine similarity was used as a measurement of category relationships [259, 262, 267], while [260] estimated the relationship by using a page-links graph to normalise the raw weight for the category edges. In addition, normalising the weight of edges can enrich the results of the clustering tags in the co-occurrence graph, e.g. [303], and can reduce the incorrect results, e.g. [304]; Also, [305] confirmed that normalising the weight improved the extraction of the semantic relationship by up to 25%. Possible methods, such as spectral clustering to identify related categories and METIS for graph partitioning would be used for the experiment.

In this thesis, to summarise the categories relationship as the similarity weight, the m-core using shared pages frequency, and the k-core quantifying the connected neighbours to weight the similarity, resulted in a few differences as discussed previously. There is room in researching the estimation of the category relationship alternatively for both cores clustering techniques; We would expect the results to be somewhat different and probably would reveal better insights of the categorised topics in Wikipedia.

Bibliography

[1] Chris Snijders, Uwe Matzat, and Ulf-Dietrich Reips. "Big Data" : big gaps of knowledge in the field of internet science. *International Journal of Internet Science*, 7(1):pages 1–5, 2012.

[2] EMC Education Services. *Data science and big data analytics*. John Wiley & Sons, 2015.

[3] Arie Croitoru, Wayant Nicole, Andrew Crooks, Jacek Radzikowski, and Stefanidis Anthony. Linking cyber and physical spaces through community detection and clustering in social media feeds. *Computers, Environment and Urban Systems*, 53:pages 47–64, 2015.

[4] Rianne Kaptein and Jaap Kamps. Exploiting the category structure of Wikipedia for entity ranking. *Artificial Intelligence*, 194:pages 111–129, 2013.

[5] Diane Joyce Cook and Lawrence Bruce Holder. *Mining graph data*. John Wiley & Sons, 2006.

[6] Wei Fan and Albert Bifet. Mining big data: current status, and forecast to the future. *The Ciation for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining*, 14(2):pages 1–5, 2013.

[7] Albert-László Barabási and Márton Pósfai. *Network science*. Cambridge University Press, 2016.

[8] Albert-László Barabási and Jennifer Frangos. *Linked: the new science of networks science of networks*. Basic Books, 2014.

[9] Mark Newman. *Networks: an introduction*. Oxford University Press, 2010.

[10] Doug Laney. 3D data management: controlling data volume, velocity and variety. *Meta Group Research Note*, 6:page 70, 2001.

[11] Ralph Schroeder and Linnet Taylor. Big data and Wikipedia research: social science knowledge across disciplinary divides. *Information, Communication & Society*, 18(9):pages 1039–1056, 2015.

[12] Olena Medelyan, David Milne, Catherine Legg, and Ian Witten. Mining meaning from Wikipedia. *International Journal of Human-Computer Studies*, 67(9):pages 716–754, 2009.

[13] Jakob Voss. Measuring Wikipedia. In *Proceedings of the 10th International Conference of the International Society for Scientometrics and Informetrics*, 2005.

[14] Luciana Salete Buriol, Carlos Castillo, Debora Donato, Stefano Leonardi, and Stefano Millozzi. Temporal analysis of the Wikigraph. In *Proceedings of the International Conference on Web Intelligence*, pages 45–51. Institute of Electrical and Electronics Engineers, 2006.

[15] Marco Gherardi, Federico Bassetti, and Marco Cosentino Lagomarsino. Law of corresponding states for open collaborations. *Physical Review E*, 93(4):page 042307, 2016.

[16] Judit Bar-Ilan and Noa Aharony. Twelve years of Wikipedia research. In *Proceedings of the Conference on Web Science*, pages 243–244. Association for Computing Machinery, 2014.

[17] Nicolas Heist and Heiko Paulheim. Uncoscale-free wireless sensor networksvering the semantics of Wikipedia categories. *Retrieved from https://arxiv.org/pdf/1906.12089 (last accessed in July 2019)*, 2019.

[18] Mike Bergman. Shaping Wikipedia into a computable knowledge base. Retrieved from http://www.mkbergman.com/1847/shaping-Wikipedia-into-a-computable\-knowledge-base/ (last accessed in July 2019), March 2015.

[19] Guo Ruiqiang and Ren Fuji. Towards the relationship between semantic web and NLP. In *Proceedings of the International Conference on the Natural Language Processing and Knowledge Engineering*, pages 1–8. Institute of Electrical and Electronics Engineers, Computer Society, 2009.

[20] Kotaro Nakayama, Hara Takahiro, and Shojiro Nishio. A thesaurus construction method from large scale web dictionaries. In *Proceedings of the 21st International Conference on Advanced Information Networking and Applications*, pages 932–939. Institute of Electrical and Electronics Engineers, 2007.

[21] Kotaro Nakayama. Extracting structured knowledge for semantic web by mining Wikipedia. In *Proceedings of the 2007 International Conference on Posters and Demonstrations- Volume 401*, International Semantic Web Conference, pages 98–99, 2008.

[22] Thomas Palomares, Youssef Ahres, Juhana Kangaspunta, and Christopher Ré. Wikipedia knowledge graph with Deepdive. In *Proceedings of the 10th Interna-*

tional Association for the Advancement of Artificial Intelligence Conference on Web and Social Media, 2016.

[23] Sadoddin Reza and Driollet Osvaldo. Mining and visualizing associations of concepts on a large-scale unstructured data. In *Proceedings of the 2nd International Conference on Big Data Computing Service and Applications (BigDataService), Institute of Electrical and Electronics Engineers*, pages 216–224, 2016.

[24] Ivan Vulić, Wim De Smet, and Marie-Francine Moens. Identifying word translations from comparable corpora using latent topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, pages 479–484, 2011.

[25] Maike Erdmann, Kotaro Nakayama, Takahiro Hara, and Shojiro Nishio. An approach for extracting bilingual terminology from Wikipedia. In *International Conference on Database Systems for Advanced Applications*, pages 380–392. Springer, 2008.

[26] Jian Hu, Lujun Fang, Yang Cao, Hua-Jun Zeng, Hua Li, Qiang Yang, and Zheng Chen. Enhancing text clustering by leveraging Wikipedia semantics. In *Proceedings of the 31st Annual International Symposium on Information Retrieval Conference on Research and Development in Information Retrieval*, pages 179–186. Association for Computing Machinery, 2008.

[27] Li Cai, Guangyou Zhou, Kang Liu, and Jun Zhao. Large-scale question classification in cQA by leveraging Wikipedia semantic knowledge. In *Proceedings of the 20th, International Conference on Information and Knowledge Management*, pages 1321–1330. Association for Computing Machinery, 2011.

[28] Pu Wang and Carlotta Domeniconi. Building semantic kernels for text classification using Wikipedia. In *Proceedings of the 14th, International Conference on Knowledge*

Discovery and Data Mining, Knowledge Discovery and Data Mining '08, pages 713–721. Association for Computing Machinery, 2008.

[29] Simone Paolo Ponzetto and Michael Strube. Deriving a large scale taxonomy from Wikipedia. In *Proceedings of Association for the Advancement of Artificial Intelligence*, volume 7, pages 1440–1445, 2007.

[30] Michael Strube and Simone Paolo Ponzetto. Wikirelate! computing semantic relatedness using Wikipedia. In *Proceedings of Association for the Advancement of Artificial Intelligence*, volume 6, pages 1419–1424, 2006.

[31] Torsten Zesch and Iryna Gurevych. Analysis of the Wikipedia category graph for NLP applications. In *Proceedings of the TextGraphs-2 Workshop, Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1–8, 2007.

[32] Mohamed Ali Hadj Taieb, Mohamed Ben Aouicha, Mohamed Tmar, and Abdelmajid Ben Hamadou. Wikipedia category graph and new intrinsic information content metric for word semantic relatedness measuring. In *Collection of Data and Knowledge Engineering*, pages 128–140. Springer, 2012.

[33] Kotaro Nakayama, Takahiro Hara, and Shojiro Nishio. Wikipedia mining for an association web thesaurus construction. In *Proceedings of the 7th International Conference on Web Information Systems Engineering*, pages 322–334. Springer, 2007.

[34] Pranam Kolari and Anupam Joshi. Web mining: research and practice. *Computing in Science Engineering, Institute of Electrical and Electronics Engineers*, 6(4):pages 49–53, 2004.

[35] Johannes Fürnkranz. Web structure mining exploiting the graph structure of the

world-wide web. *Oesterreichische Gesellschaft fuer Artificial Intelligence Journal*, 21(2):pages 17–26, 2002. Special Issue on Web Mining.

[36] Soumen Chakrabarti. Data mining for hypertext: a tutorial survey. *The Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining, Explor. Newsl.*, 1(2):pages 1–11, 2000.

[37] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2014.

[38] Behnam Rahdari and Peter Brusilovsky. Building a knowledge graph for recommending experts. Retrieved from http://di2kg.inf.uniroma3.it/papers/DI2KG_paper_2.pdf, (last accessed in July 2019).

[39] Heba Ismail and Boumediene Belkhouche. Evaluating the impact of personalized content recommendations on informal learning from Wikipedia. In *Proceedings of 2019 Institute of Electrical and Electronics Engineers, Global Engineering Education Conference*, pages 943–952. Institute of Electrical and Electronics Engineers, 2019.

[40] Ricardo Baeza-Yates. Bias on the Web. *Communications of the Association for Computing Machinery*, 61(6):pages 54–61, 2018.

[41] Christoph Hube. Bias in Wikipedia. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 717–721. International World Wide Web Conferences Steering Committee, 2017.

[42] Claudia Wagner, David Garcia, Mohsen Jadidi, and Markus Strohmaier. It's a man's Wikipedia? assessing gender inequality in an online encyclopedia. In *Proceedings of the 9th International Association for the Advancement of Artificial Intelligence Conference on Web and Social Media*, 2015.

[43] Joseph Reagle and Lauren Rhue. Gender bias in Wikipedia and Britannica. *International Journal of Communication*, 5:page 21, 2011.

[44] Eduardo Graells-Garrido, Mounia Lalmas, and Filippo Menczer. First women, second sex: Gender bias in Wikipedia. In *Proceedings of the 26th Association for Computing Machinery Conference on Hypertext & Social Media*, pages 165–174. Association for Computing Machinery, 2015.

[45] Wikipedia. Information about Wikipedia. Provided at https://en.Wikipedia.org/wiki/Wikipedia:About (last accessed in July), 2019.

[46] Wikipedia. Core of content policies of Wikipedia. Provided at https://en.wikipedia.org/wiki/Wikipedia:Core_content_policies (last accessed in July), 2019.

[47] Wikipedia. Content criteria of Wikipedia. Provided at https://en.wikipedia.org/wiki/Wikipedia:About#Wikipedia_content_criteria (last accessed in July), 2019.

[48] John Broughton. *Wikipedia: the missing manual*. Missing Manual. O'Reilly Media, 2008.

[49] Denise Anthony, Sean Smith, and Timothy Williamson. Reputation and reliability in collective goods the case of the online encyclopedia Wikipedia. *Rationality and Society*, 21:pages 283–306, 2009.

[50] Wikipedia. Categorisation guidance of Wikipedia. Provided at https://en.Wikipedia.org/wiki/Wikipedia:Categorization (last accessed in July), 2019.

[51] Wikipedia. About Wikipedia's categories. Provided at https://en.wikipedia.org/wiki/Wikipedia:FAQ/Categories (last accessed in July), 2019.

[52] Wikipedia. About Wikipedia's categorisation. Provided at https://en.wikipedia.org/wiki/Wikipedia:FAQ/Categorization (last accessed in July), 2019.

[53] Aniket Kittur, EH Chi, and Bongwon Suh. What's in Wikipedia?: mapping topics and conflict using socially annotated category structure. In *Proceedings of the Special Interest Group on Computer-Human Interaction Conference on Human Factors in Computing Systems*, pages 1509–1512. Association for Computing Machinery, 2009.

[54] Ekaterina Chernyak and Boris Mirkin. A method for refining a taxonomy by using annotated suffix trees and Wikipedia resources. *Procedia Computer Science*, 31:pages 193–200, 2014.

[55] Krzysztof Suchecki, Alkim Almila Akdag Salah, Cheng Gao, and Andrea Scharnhorst. Evolution of Wikipedia's category structure. *Advances in Complex Systems*, 15:page 1250068, 2012.

[56] Maciej Janik and Krys Kochut. Wikipedia in action: Ontological knowledge in text categorization. In *Proceedings of the International Conference 2008, Institute of Electrical and Electronics Engineers on Semantic Computing*, pages 268–275. Institute of Electrical and Electronics Engineers, 2008.

[57] David Milne, Olena Medelyan, and Ian Witten. Mining domain-specific thesauri from Wikipedia: a case study. In *Proceedings of the 2006 International Conference on Web Intelligence*, pages 442–448. Institute of Electrical and Electronics Engineers, Computer Society, 2006.

[58] Robert Biuk-Aghai, Cheong-Iao Pang, and Yain-Whar Si. Visualizing large-scale human collaboration in Wikipedia. *Future Generation Computer Systems*, 31:pages 120–133, 2014.

[59] Jakob Voss. Collaborative thesaurus tagging the Wikipedia way. Computer Science, Arxiv e-prints, Retrieved from https://arxiv.org/pdf/cs/0604036 (last accessed in July 2019), 2006.

[60] Elin Jacob. Classification and categorization: a difference that makes a difference. *Trends*, 52(3):pages 515–540, 2004.

[61] Scott Golder and Bernardo HubermanBernardo. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):pages 198–208, 2006.

[62] Paul Heymann and Hector Garcia-Molina. Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical report, Stanford InfoLab, April 2006.

[63] Almila Akdag Salah, Cheng Gao, Krzysztof Suchecki, and Andrea Scharnhorst. Need to categorize: A comparative look at the categories of universal decimal classification system and Wikipedia. *Leonardo*, 45(1):pages 84–85, 2012.

[64] Wikipedia. Policies and guidelines of Wikipedia. Provided at https://en.Wikipedia.org/wiki/Wikipedia:Policies_and_guidelines (last accessed in July) 2019.

[65] Thidawan Klaysri, Trevor Fenner, Oded Lachish, Mark Levene, and Panagiotis Papapetrou. Analysis of cluster structure in large-scale English Wikipedia category networks. In *Proceedings of the International Symposium on Intelligent Data Analysis*, pages 261–272. Springer, 2013.

[66] Ramakrishna Bairi, Mark Carman, and Ganesh Ramakrishnan. On the evolution of Wikipedia: dynamics of categories and articles. In *Proceedings of the 9th International Conference on Web and Social Media*. Association for the Advancement of Artificial Intelligence, 2015.

[67] Simone Paolo Ponzetto and Michael Strube. Taxonomy induction based on a collaboratively built knowledge repository. *Artificial Intelligence*, 175:pages 1737–1756, 2011.

[68] Tiziano Flati, Daniele Vannella, Tommaso Pasini, and Roberto Navigli. Two is bigger (and better) than one: the Wikipedia bitaxonomy project. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistic*, pages 945–955, 2014.

[69] Suchanek Fabian, Kasneci Gjergji, and Weikum Gerhard. Yago: a core of semantic knowledge unifying Wordnet and Wikipedia. In *Proceedings of the 16th International World Wide Web Conference*, pages 697–706, 2007.

[70] Simone Paolo Ponzetto and Michael Strube. Wikitaxonomy: a large scale knowledge resource. In *Proceedings of European Conference on Artificial Intelligence*, volume 178, pages 751–752, 2008.

[71] Cäcilia Zirn, Vivi Nastase, and Michael Strube. Distinguishing between instances and classes in the Wikipedia taxonomy. In *Proceedings European Semantic Web Conference*, pages 376–387. Springer, 2008.

[72] Vijay Gadepally and Jeremy Kepner. Using a power law distribution to describe big data. In *Proceedings of High Performance Extreme Computing Conference*, pages 1–5, 2015.

[73] Colaiori Francesca Buriol Luciana Donato Debora Leonardi Stefano Capocci Andrea, Servedio Vito and Guido Caldarelli. Preferential attachment in the growth of social networks: the internet encyclopedia Wikipedia. *Physical Review E*, 74(3):page 036116, 2006.

[74] Vinko Zlatić, Miran Božičević, Hrvoje Štefančić, and Mladen Domazet. Wikipedias: collaborative web-based encyclopedias as complex networks. *Physical Review E*, 74:page 016115, 2006.

[75] David Laniado, Riccardo Tasso, Yana Volkovich, and Andreas Kaltenbrunner. When the Wikipedians talk: network and tree structure of Wikipedia discussion pages. pages 177–84, 2011.

[76] Vivek Kumar Singh and Ramesh Jain. Structural analysis of the emerging event-web. In *Proceedings of the 19th International Conference on World Wide Web*, pages 1183–1184. Association for Computing Machinery, 2010.

[77] Alessandra Sala, Haitao Zheng, Ben Zhao, Sabrina Gaito, and Gian Paolo Rossi. Brief announcement: revisiting the power-law degree distribution for social graph analysis. In *Proceedings of the 29th Association for Computing Machinery, Symposium on*, Principles of Distributed Computing '10, pages 400–401.

[78] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):pages 509–512, 1999.

[79] Yuanwei Liu, Lifeng Wang, Syed Ali Raza Zaidi, Maged Elkashlan, and Trung Q Duong. Secure D2D communication in large-scale cognitive cellular networks: a wireless power transfer model. *Institute of Electrical and Electronics Engineers, Transactions on Communications*, 64(1):pages 329–342, 2016.

[80] Soon-Hyung Yook, Hawoong Jeong, and Albert-László Barabási. Modeling the internet's large-scale topology. *The National Academy of Sciences*, 99(21):pages 13382–13386, 2002.

[81] Ergin Yilmaz, Veli Baysal, Matjaž Perc, and Mahmut Ozer. Enhancement of pace-maker induced stochastic resonance by an autapse in a scale-free neuronal network. *Science China Technological Sciences*, 59(3):pages 364–370, 2016.

[82] Paolo Massobrio, Valentina Pasquale, and Sergio Martinoia. Self-organized criticality

in cortical assemblies occurs in concurrent scale-free and small-world networks. *Scientific reports*, 5:page 10578, 2015.

[83] Tie Qiu, Aoyang Zhao, Feng Xia, Weisheng Si, Dapeng Oliver Wu, Tie Qiu, Aoyang Zhao, Feng Xia, Weisheng Si, and Dapeng Oliver Wu. ROSE: Robustness strategy for scale-free wireless sensor networks. *Institute of Electrical and Electronics Engineers/ Association for Computing Machinery, Transactions on Networking*, 25(5):pages 2944–2959, 2017.

[84] Xingzhao Peng, Hong Yao, Jun Du, Zhe Wang, and Chao Ding. Invulnerability of scale-free network against critical node failures based on a renewed cascading failure model. *Physica A: Statistical Mechanics and its Applications*, 421:pages 69–77, 2015.

[85] Lada Adamic and Bernardo Huberman. Zipf's law and the internet. *Glottometrics*, 3:pages 143–150, 2002.

[86] Aaron Clauset, Cosma Rohilla Shalizi, and Mark Newman. Power-law distributions in empirical data. *Society for Industrial and Applied Mathematics Review*, 51(4):pages 661–703, 2009.

[87] Mark Newman. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, 46(5):pages 323–351, 2005.

[88] Lada A Adamic. Zipf, power-laws, and pareto-a ranking tutorial. *Xerox Palo Alto Research Center, Palo Alto, CA*, 2000.

[89] Willem Robert Van Hage, Thomas Ploeger, and Jesper Hoeksema. Number frequency on the Web. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 571–572, 2014.

[90] Jérôme Kunegis, Marcel Blattner, and Christine Moser. Preferential attachment in online networks: Measurement and explanations. In *Proceedings of the 5th annual web science conference*, pages 205–214. Association for Computing Machinery, 2013.

[91] Alexei Vázquez. Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. *Physical Review E*, 67(5):page 056104, 2003.

[92] Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic spreading in scale-free networks. *Physical Review Letters*, 86:pages 3200–3203, April 2001.

[93] Adolfo Paolo Masucci, Alkiviadis Kalampokis, Victor Martínez Eguíluz, and Emilio Hernández-García. Wikipedia information flow analysis reveals the scale-free architecture of the semantic space. *PLoS one*, 6(2):pages 1–7, 2011.

[94] Petter Holme and Beom Jun Kim. Growing scale-free networks with tunable clustering. *Physical Review E*, 65:page 026107, Jan 2002.

[95] Albert-László Barabási. Scale-free networks: A decade and beyond. *Science*, 325(5939):pages 412–413, 2009.

[96] Albert-László Barabási and Eric Bonabeau. Scale-free networks. *Scientific american*, 288(5):pages 60–69, 2003.

[97] Mark Newman. The structure of scientific collaboration networks. *The national academy of sciences*, 98(2):pages 404–409, 2001.

[98] Deepayan Chakrabarti and Christos Faloutsos. Graph mining: Laws, generators, and algorithms. *Association for Computing Machinery Computing Surveys*, 38(1):2, 2006.

[99] Mark Levene. *An introduction to search engines and web navigation*. John Wiley & Sons, 2011.

[100] Deepayan Chakrabarti and Christos Faloutsos. *Graph Mining: laws, tools, and case studies*. Morgan & Claypool Publishers, 2012.

[101] Luıs A Nunes Amaral, Antonio Scala, Marc Barthelemy, and Harry Eugene Stanley Stanley. Classes of small-world networks. *The national academy of sciences*, 97(21):pages 11149–11152, 2000.

[102] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1), 2002.

[103] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. 29(4):pages 251–262, 1999.

[104] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. *Computer networks*, 33(1-6):pages 309–320, 2000.

[105] Caroline S Wagner and Loet Leydesdorff. Network structure, self-organization, and the growth of international collaboration in science. *Research policy*, 34(10):pages 1608–1618, 2005.

[106] Lawrence Gostin and Lindsay Wiley. *Public health law: power, duty, restraint*. Univ of California Press, 2016.

[107] Bater Makhabel. *Learning data mining with R*. Community experience distilled. Packt Publishing, Limited, 2015.

[108] Rui Xu and Don Wunsch. *Clustering*. Institute of Electrical and Electronics Engineers, Press Series on Computational Intelligence. Wiley, 2008.

[109] Max Bramer. *Principles of Data Mining*. Undergraduate Topics in Computer Science. Springer, 2007.

[110] Pavel Berkhin. *A Survey of Clustering Data Mining Techniques*, pages 25–71. Springer Berlin Heidelberg, 2006.

[111] Nagiza Samatova, William Hendrix, John Jenkins, Kanchana Padmanabhan, and Arpan Chakraborty. *Practical graph mining with R*. Chapman & Hall/CRC, 2013.

[112] Boris Mirkin. *Clustering: a data recovery approach, second edition*. Chapman & Hall/CRC Computer Science & Data Analysis. CRC Press, 2012.

[113] Francis R Bach and Michael I Jordan. Learning spectral clustering. In *Advances in neural information processing systems*, pages 305–312, 2004.

[114] Maurizio Filippone, Francesco Camastra, Francesco Masulli, and Stefano Rovetta. A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41(1):pages 176–190, 2008.

[115] Srinivasan Parthasarathy, Shirish Tatikonda, and Duygu Ucar. *A Survey of Graph Mining Techniques for Biological Datasets*, pages 547–580. Springer US, 2010.

[116] Charu Aggarwal and Haixun Wang. *Graph data management and mining: a survey of algorithms and applications*, pages 13–68. Springer US, 2010.

[117] Lei Tang and Huan Liu. Graph mining applications to social network analysis. In *Collection of Managing and Mining Graph Data*, pages 487–513. Springer, 2010.

[118] Saif Ur Rehman, Asmat Ullah Khan, and Simon Fong. Graph mining: a survey of graph mining techniques. In *Proceedings of the 7th International Conference on Digital Information Management*, pages 88–92. Institute of Electrical and Electronics Engineers, 2012.

[119] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):pages 27–64, 2007.

[120] Luca Zanetti. *Algorithms for partitioning well-clustered graphs*. PhD thesis, University of Bristol, 2018.

[121] Tapas Kanungo, David Mount, Nathan Netanyahu, Christine Piatko, Ruth Silverman, and Angela Wu. An efficient k-means clustering algorithm: Analysis and implementation. *Institute of Electrical and Electronics Engineers, Transactions on Pattern Analysis and Machine Intelligence*, (7):pages 881–892, 2002.

[122] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *Proceedings of the International Conference on Machine Learning*, volume 1, pages 577–584, 2001.

[123] Aristidis Likas, Nikos Vlassis, and Jakob Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36(2):pages 451–461, 2003. Biometrics.

[124] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th International Conference on World Wide Web*, pages 1177–1178. Association for Computing Machinery, 2010.

[125] Weizhong Zhao, Huifang Ma, and Qing He. Parallel k-means clustering based on mapreduce. In *Proceedings of the International Conference on Cloud Computing, Institute of Electrical and Electronics Engineers*, pages 674–679. Springer, 2009.

[126] Anil Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):pages 651–666, 2010.

[127] Xin Jin and Jiawei Han. K-medoids clustering. In *Collection of Encyclopedia of Machine Learning and Data Mining*, pages 1–3. Springer, 2016.

[128] T Velmurugan and T Santhanam. Computational complexity between k-means and k-medoids clustering algorithms for normal and uniform distributions of data points. *Journal of computer science*, 6(3):page 363, 2010.

[129] Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, 36(2):pages 3336–3341, 2009.

[130] Seema Bandyopadhyay and Edward Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *Proceedings of the 21th Annual Joint Conference of the Institute of Electrical and Electronics Engineers Computer and Communications Societies*, volume 3, pages 1713–1723, 2003.

[131] Deng Cai, Xiaofei He, Zhiwei Li, Wei-Ying Ma, and Ji-Rong Wen. Hierarchical clustering of WWW image search results using visual, textual and link information. In *Proceedings of the 12th Annual International Conference on Multimedia*, pages 952–959. Association for Computing Machinery, 2004.

[132] Andriy Shepitsen, Jonathan Gemmell, Bamshad Mobasher, and Robin Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2008 Association for Computing Machinery Conference on Recommender Systems*, pages 259–266. Association for Computing Machinery, 2008.

[133] Christopher Brooks and Nancy Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *Proceedings of the 15th International Conference on World Wide Web*, pages 625–632. Association for Computing Machinery, 2006.

[134] Gerasimos Spanakis, Georgios Siolas, and Andreas Stafylopatis. Exploiting Wikipedia knowledge for conceptual hierarchical clustering of documents. *The Computer Journal*, 55(3):pages 299–312, 2012.

[135] Guoyu Tang, Yunqing Xia, Weizhi Wang, Raymond Lau, and Fang Zheng. Clustering tweets using Wikipedia concepts. In *Proceedings of The International Conference on Language Resources and Evaluation*, pages 2262–2267, 2014.

[136] Ying Zhao and George Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the 11th International Conference on Information and Knowledge Management*, pages 515–524. Association for Computing Machinery, 2002.

[137] Ying Zhao, George Karypis, and Usama Fayyad. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10:pages 141–168, 2005.

[138] David Martin Lydon-Staley, Dale Zhou, Ann Sizemore Blevins, Perry Zurn, and Danielle S Bassett. Hunters, busybodies, and the knowledge network building associated with curiosity. 2019.

[139] Preeti Bhargava, Nemanja Spasojevic, Sarah Ellinger, Adithya Rao, Abhinand Menon, Saul Fuhrmann, and Guoning Hu. Learning to map Wikidata entities to predefined topics. In *Companion Proceedings of the 2019 World Wide Web Conference*, pages 1194–1202. Association for Computing Machinery, 2019.

[140] Angel Conde, Mikel Larrañaga, Ana Arruarte, and Jon A Elorriaga. A combined approach for eliciting relationships for educational ontologies using general-purpose knowledge bases. *Institute of Electrical and Electronics Engineers Access*, 7:pages 48339–48355, 2019.

[141] Miriam Redi, Besnik Fetahu, Jonathan Morgan, and Dario Taraborelli. Citation needed: A taxonomy and algorithmic assessment of Wikipedia's verifiability. In *Proceedings of the World Wide Web Conference*, pages 1567–1578. Association for Computing Machinery, 2019.

[142] Pu Wang, Jian Hu, Hua-Jun Zeng, and Zheng Chen. Using Wikipedia knowledge to

improve text classification. *Knowledge and Information Systems*, 19:pages 265–281, 2009.

[143] Guangyou Zhou, Yang Liu, Fang Liu, Daojian Zeng, and Jun Zhao. Improving question retrieval in community question answering using world knowledge. In *Proceedings of the International Joint Committee on Artificial Intelligence*, volume 13, pages 2239–2245, 2013.

[144] Daniil Mirylenka and Andrea Passerini. Navigating the topical structure of academic search results via the Wikipedia category network. In *Proceedings of the 22nd International Conference on Information & Knowledge Management*, pages 891–896. Association for Computing Machinery, 2013.

[145] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):pages 395–416, 2007.

[146] E Krisna, Alhadi Bustamam, and Kiki Ariyanti Sugeng. Clustering protein-protein interaction data with spectral clustering and fuzzy random walk. In *Journal of Physics: Conference Series*, volume 1211, page 012027. IOP Publishing, 2019.

[147] Maria CV Nascimento and Andre CPLF De Carvalho. Spectral methods for graph clustering-a survey. *European Journal of Operational Research*, 211(2):pages 221–231, 2011.

[148] Benjamin Auffarth. Spectral graph clustering. *Universitat de Barcelona, course report for Technicas Avanzadas de Aprendizaj, at Universitat Politecnica de Catalunya*, 2007.

[149] Zakariyaa Ait El Mouden, Abdeslam Jakimi, and Moha Hajar. An application of spectral clustering approach to detect communities in data modeled by graphs. In *Proceedings of the 2nd International Conference on Networking, Information Systems & Security*, page 4. Association for Computing Machinery, 2019.

[150] Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. Graph clustering based on structural/attribute similarities. *Proceedings of the Very Large Data Base Endowment Inc.*, 2(1):pages 718–729, 2009.

[151] Christos Giatsidis, Dimitrios Thilikos, and Michalis Vazirgiannis. Evaluating cooperation in communities with the k-core structure. In *Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining*, pages 87–93, 2011.

[152] Enis Arslan, Selim Akyokuş, and Murat Can Ganiz. An application of community discovery in academical social networks. In *Proceedings of the 2013 Institute of Electrical and Electronics Engineers on Innovations in Intelligent Systems and Applications*, pages 1–5, 2013.

[153] Mohammed Al-Taie and Seifedine Kadry. Applying social network analysis to analyze a web-based community. *Editorial Preface*, 3(2), 2012.

[154] Nasrullah Memon, Kim Kristoffersen, David Hicks, and Henrik Legind Larsen. Detecting critical regions in covert networks: a case study of 9/11 terrorists networks. In *Proceedings of the 2nd International Conference on Availability, Reliability and Security, 2007*, pages 861–870. Institute of Electrical and Electronic Engineers, 2007.

[155] Nasrullah Memon, Uffe Kock Wiil, Pir Abdul Rasool Qureshi, and Panagiotis Karampelas. *Retracted: exploring the evolution of terrorist network*. Springer, 2011.

[156] Alvin Chin and Mark Chignell. Identifying active subgroups in online communities. In *Proceedings of the 2007 Conference of the Center for Advanced Studies on Collaborative Research*, pages 280–283. IBM Corp., 2007.

[157] Zhuqi Miao and Balabhaskar Balasundaram. Cluster detection in large-scale social networks using k-plexes. In *Proceedings of the Annual Institute of Industrial Engineers Conference*, page 1. Institute of Industrial Engineers-Publisher, 2012.

[158] Christina Pikas. Detecting communities in science blogs. In *Proceedings of the 4th International Conference on eScience*, pages 95–102. Institute of Electrical and Electronic Engineers, 2008.

[159] Dongsheng Duan, Yuhua Li, Ruixuan Li, and Zhengding Lu. Incremental *K*-clique clustering in dynamic social networks. *Artificial Intelligence Review*, 38(2):pages 129–147, 2012.

[160] Stanley Wasserman and Katherine Faust. *Social network analysis: methods and applications*, volume 8. Cambridge university press, 1994.

[161] John Scott and Peter Carrington. *The SAGE handbook of social network analysis*. SAGE Publications, 2011.

[162] David Knoke and Song Yang. *Social network analysis*, volume 154 of *Quantitative Applications in the Social Sciences*. SAGE Publications, 2nd edition, 2008.

[163] Mark Newman. The structure and function of complex networks. *Society for Industrial and Applied Mathematics Review*, 45(2):pages 167–256, 2003.

[164] David Easley and Jon Kleinberg. *Networks, crowds, and markets: reasoning about a highly connected world*. Cambridge University Press, 2010.

[165] Charles Kadushin. *Understanding social networks: theories, concepts, and findings*. Oxford University Press, 2011.

[166] Alain Degenne and Michel Forsé. *Introducing social networks*. SAGE Publications, 1999.

[167] Jeroen Bruggeman. *Social networks: an introduction*. Routledge, 2008.

[168] Takashi Iba, Keiichi Nemoto, Bernd Peters, and Peter Gloor. Analyzing the creative editing behavior of Wikipedia editors: through dynamic social network analysis. *Procedia-Social and Behavioral Sciences*, 2(4):pages 6441–6456, 2010.

[169] Mark Granovetter. The strength of weak ties. *American Journal of Sociology*, pages 1360–1380, 1973.

[170] John Scott. *Social network analysis: a handbook*. SAGE Publications, second. edition, 2000.

[171] Peter Carrington, John Scott, and Stanley Wasserman, editors. *Models and methods in social network analysis*. Structural analysis in the social sciences. Cambridge University Press, 2005.

[172] Qun Chen, Li You, Zhanhuai Li, and Zachary Ives. Parallelizing clique and quasi-clique detection over graph data. 2014.

[173] Balabhaskar Balasundaram, Sergiy Butenko, and Illya Hicks. Clique relaxations in social network analysis: the maximum k-plex problem. *Operations Research*, 59(1):pages 133–142, 2011.

[174] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):pages 814–818, 2005.

[175] Charalampos Tsourakakis. The k-clique densest subgraph problem. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1122–1132. Association for Computing Machinery, 2015.

[176] Robert Mokken. Cliques, clubs and clans. *Quality and Quantity*, 13(2):pages 161–173, 1979.

[177] Conrad Lee, Aaron McDaid, Fergal Reid, and Neil Hurley. Detecting highly overlapping community structure by greedy clique expansion. In *Proceedings of the 4th Workshop on Social Network Mining and Analysis held in Conjunction with the International Conference on Knowledge Discovery and Data Mining*, pages 33–42, 2010.

[178] Ayman Alhelbawy and Robert Gaizauskas. Collective named entity disambiguation using graph ranking and clique partitioning approaches. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 1544–1555, 2014.

[179] Yotaro Watanabe, Masayuki Asahara, and Yuji Matsumoto. A graph-based approach to named entity categorization in Wikipedia using conditional random fields. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing and Computational Natural Language Learning*, pages 649–657, 2007.

[180] Łukasz Bolikowski. Scale-free topology of the interlanguage links in Wikipedia. *Retrieved from https://arxiv.org/pdf/0904.0564 (last accessed in July 2019)*, 2009.

[181] I-Chin Wu and Yi-Sheng Lin. WNavis:navigating Wikipedia semantically with an SNA-based summarization technique. *Decision Support Systems*, 54(1):pages 46–62, 2012.

[182] Amir Jadidinejad and Fariborz Mahmoudi Mahmoudi. Clique-based semantic kernel with application to semantic relatedness. *Natural Language Engineering*, 21(05):pages 725–742, 2015.

[183] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. collective annotation of Wikipedia entities in web text. In *Proceedings of the 15th International conference Special Interest Group on Knowledge Discovery and Data Mining*, pages 457–466. Association for Computing Machinery, 2009.

[184] WenKe Yin, Ming Zhu, and TianHao Chen. Domain thesaurus construction from Wikipedia. In *Proceedings of the International Conference on Computer Networks and Communications Engineering*, pages 87–92, 2013.

[185] Zhaohui Wu and Lee Giles. Sense-aware semantic analysis: a multi-prototype word representation model using Wikipedia. In *Proceedings of Association for the Advancement of Artificial Intelligence*, pages 2188–2194, 2015.

[186] Tim Evans. Clique graphs and overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment*, 2010(12):page 12037, 2010.

[187] Santo Fortunato and Claudio Castellano. Community structure in graphs. In *Collection of Computational Complexity*, pages 490–512. Springer, 2012.

[188] Alvin Chin and Mark Chignell. Identifying subcommunities using cohesive subgroups in social hypertext. In *Proceedings of the 18th Conference on Hypertext and Hypermedia*, pages 175–178. Association for Computing Machinery, 2007.

[189] Stephen Seidman. Network structure and minimum degree. *Social networks*, 5(3):pages 269–287, 1983.

[190] Wissam Khaouid, Marina Barsky, Venkatesh Srinivasan, and Alex Thomo. K-core decomposition of large networks on a single pc. *Proceedings of the VLDB Endowment*, 9(1):pages 13–23, 2015.

[191] Yi-Hui Lin, De-Nian Yang, and Wen-Tsuen Chen. Privacy-preserving dense subgraph discovery in mobile social networks. In *Proceedings of the 2015 Institute of Electrical and Electronics Engineers on Global Communications Conference*, pages 1–7, 2015.

[192] Patrick Doreian and Katherine Woodard. Defining and locating cores and boundaries of social networks. *Social Networks*, 16(4):pages 267–293, 1994.

[193] Paul Jakma, Marcin Orczyk, Colin Perkins, and Marwan Fayed. Distributed k-core decomposition of dynamic graphs. In *Proceedings of the 2012 Conference on CoNEXT Student Workshop*, pages 39–40. Association for Computing Machinery, 2012.

[194] Norihiko Kamakura, Hiroki Takahashi, Kensuke Nakamura, Shigehiko Kanaya, and Altaf-Ul-Amin. Protein function prediction based on k-cores of interaction networks. In *Proceedings of the 2010 International Conference on Bioinformatics and Biomedical Technology*, pages 211–215, 2010.

[195] Fahad Saeed, Jason Hoffert, and Mark Knepper. A high performance algorithm for clustering of large-scale protein mass spectrometry data using multi-core architectures. In *Proceedings of the 2013 International Conference on Advances in Social Networks Analysis and Mining*, pages 923–930, 2013.

[196] Yizong, Chen Lu, and Nan Wang. Local k-core clustering for gene networks. In *Proceedings of the 2013 Institute of Electrical and Electronics Engineer on Bioinformatics and Biomedicine*, pages 9–15, 2013.

[197] Vladimir Batagelj and Andrej Mrvar. Pajek-program for large network analysis. *Connections*, 21(2):pages 47–57, 1998.

[198] Hao Huang, Yunjun Gao, Kevin Chiew, Qinming He, and Baihua Zheng. Unsupervised analysis of top-k core members in poly-relational networks. *Expert Systems with Applications*, 41(13):pages 5689–5701, 2014.

[199] Alvin Chin and Mark Chignell. A social hypertext model for finding community in blogs. In *Proceedings of the 17th Conference on Hypertext and Hypermedia*, pages 11–22. Association for Computing Machinery, 2006.

[200] Vladimir Batagelj and Matjaz Zaversnik. An O(m) algorithm for cores decomposition of networks. Computer Science, Arxiv e-prints, Retrieved from https://arxiv.org/pdf/cs/0310049 (last accessed in July 2019), 2003.

[201] Shai Carmi, Shlomo Havlin, Scott Kirkpatrick, Yuval Shavitt, and Eran Shir. a model of internet topology using k-shell decomposition. *Proceedings of the National Academy of Sciences*, 104(27):pages 11150–11154, 2007.

[202] Xiaoli Li, Min Wu, Chee-Keong Kwoh, and See-Kiong Ng. Computational approaches for detecting protein complexes from protein interaction networks: a survey. *BioMed Central Genomics*, 11:page S3, 2010.

[203] Xuan He, Hai Zhao, Wei Cai, Guang-Guang Li, and Fan-Dong Pei. Analyzing the structure of earthquake network by k-core decomposition. *Physica A: Statistical Mechanics and its Applications*, 421:pages 34–43, 2015.

[204] François Rousseau, Emmanouil Kiagias, and Michalis Vazirgiannis. Text categorization as a graph classification problem. In *Proceedings of the Conference of the Association for Computational Linguistics*, volume 15, page 107, 2015.

[205] Ahmet Erdem Saríyüce, Buğra Gedik, Gabriela Jacques-Silva, Kun-Lung Wu, and Ümit V Çatalyürek. Streaming algorithms for k-core decomposition. *Very Large Data Base Endowment*, 6(6):pages 433–444, 2013.

[206] James Cheng, Yiping Ke, Shumo Chu, and Tamer Özsu. Efficient core decomposition in massive networks. In *Proceedings of the 27th International Conference on Data Engineering*, pages 51–62. Institute of Electrical and Electronic Engineers, 2011.

[207] Rong-Hua Li, Jeffrey Xu Yu, and Rui Mao. Efficient core maintenance in large dynamic graphs. *Institute of Electrical and Electronic Engineers Transactions on Knowledge and Data Engineering*, 26(10):pages 2453–2465, 2014.

[208] Fragkiskos Malliaros and Michalis Vazirgiannis. To stay or not to stay: modeling engagement dynamics in social graphs. In *Proceedings of the 22nd, International Conference on Information & Knowledge Management*, pages 469–478. Association for Computing Machinery, 2013.

[209] Marius Eidsaa and Eivind Almaas. S-core network decomposition: A generalization of k-core analysis to weighted networks. *Physical Review E*, 88(6):page 062819, 2013.

[210] Alberto Montresor, Francesco De Pellegrini, and Daniele Miorandi. Distributed k-core decomposition. *Transactions on Parallel and Distributed Systems, Institute Of Electrical And Electronics Engineers*, 24(2):pages 288–300, Feb 2013.

[211] George Barnett. *Encyclopedia of social networks*. SAGE Publications, 2011.

[212] John Beaumont and Anthony Gatrell. An introduction to q-analysis. *Concept and Techiques in Modern Geography No 34*, 1986.

[213] Ron Atkin. *Mathematical structure in human affairs*. Crane, Russak, 1975.

[214] John Scott. Capitalist property and financial power. *Brighton: Wheatsheaf*, 1986.

[215] Lucien Duckstein and Steven Nobe. Q-analysis for modeling and decision making. *European Journal of Operational Research*, 103(3):pages 411–425, 1997.

[216] Linton Freeman. Q-analysis and the structure of friendship networks. *International Journal of Man-Machine Studies*, 12:pages 367–378, 1980.

[217] Thomas Jacobson, David Fusani, and Wenjie Yan. Q-analysis of user-database interaction. *International Journal of Man-Machine Studies*, 38:pages 787–803, 1993.

[218] Jiang He and Hosein Fallah. Dynamics of inventors' network and growth of geographic clusters: evidence from telecommunications industry in NJ & TX. In *Proceedings of Portland International Conference on Management of Engineering*

and Technology on Technology Management for the Global Future, volume 2, pages 815–825. Institute of Electrical and Electronic Engineers, 2006.

[219] Xiao Liu and Jianmei Yang. Network analysis and rule mining of export market structure evolvement. In *Proceedings of the International Conference on Systems Man and Cybernetics*, pages 713–719. Institute of Electrical and Electronic Engineers, 2010.

[220] Christian Del Rosso. Comprehend and analyze knowledge networks to improve software evolution. *Journal of Software Maintenance and Evolution: Research and Practice*, 21(3):pages 189–215, 2009.

[221] Daniel Rodríguez, Miguel Ángel Sicilia, Salvador Sánchez-Alonso, Leonardo Lezcano, and Elena García-Barriocanal. Exploring affiliation network models as a collaborative filtering mechanism in e-learning. *Interactive Learning Environments*, 19(4):pages 317–331, 2011.

[222] Leonardo Lezcano, Salvador Sánchez-Alonso, and Miguel-Angel Sicilia. Associating clinical archetypes through UMLs metathesaurus term clusters. *Journal of Medical Systems*, 36(3):pages 1249–1258, 2012.

[223] Thomas David and Gerarda Westerhuis. *The power of corporate networks: A comparative and historical perspective*, volume 26. Routledge, 2014.

[224] Falk Strotebeck. What is behind the structure of regional networks in the German biotechnology industry? 2010.

[225] Tomás Isakowitz, Arnold Kamis, and Marios Koufaris. Extending the capabilities of RMM: Russian dolls and hypertext. In *Proceedings of the 30th Hawaii International Conference on System Sciences*, volume 6, pages 177–186. Institute of Electrical and Electronic Engineers, 1997.

[226] Flavius Frasincar, Geert Jan Houben, and Richard Vdovjak. An RMM-based methodology for hypermedia presentation design. In *Proceedings of the East European Conference on Advances in Databases and Information Systems*, pages 323–337. Springer, 2001.

[227] Tomas Isakowitz, Arnold Kamis, and Marios Koufaris. The extended RMM methodology for web publishing. *New York University*, 1998.

[228] Carlos Miguel Tobar and Ivan Luiz Marques Ricarte. Towards a categorization of hypermedia data models. In *Proceedings of Multimedia Modeling*, pages 79–95. Singapura: World Scientific, 1999.

[229] Tomás Isakowitz, Edward Stohr, and Papyrus Balasubramanian. RMM: a methodology for structured hypermedia design. *Communications of the Association for Computing Machinery*, 38(8):pages 34–44, 1995.

[230] Tomas Isakowitz, Arnold Kamis, and Marios Koufaris. Reconciling top-down and bottom-up design approaches in RMM. *Association for Computing Machinery, Special Interest Group on Management Information Systems-Database*, 29(4):pages 58–67, 1998.

[231] Philip Leifeld and Sebastian Haunss. Political discourse networks and the conflict over software patents in Europe. *European Journal of Political Research*, 51(3):pages 382–409, 2012.

[232] Preston Aldrich. Diffusion limited aggregation and the fractal evolution of gene promoter networks. *Network Biology*, 1(2):page 99, 2011.

[233] Qi Yu, Hongfang Shao, and Zhiguang Duan. Research groups of oncology co-authorship network in China. *Scientometrics*, 89(2):pages 553–567, 2011.

[234] Qi Yu, Hongfang Shao, and Zhiguang Duan. The research collaboration in Chinese cardiology and cardiovasology field. *International Journal of Cardiology*, 167(3):pages 786–791, 2013.

[235] Ce Zhang. *DeepDive: a data management system for automatic knowledge base construction*. PhD thesis, The University of Wisconsin-Madison, 2015.

[236] Dan Jurafsky and James Martin. *Speech and language processing*, volume 3. Pearson London, 2014.

[237] Ching Suen. N-gram statistics for natural language understanding and text processing. *Institute of Electrical and Electronics Engineers, Transactions on Pattern Analysis and Machine Intelligence*, (2):pages 164–172, 1979.

[238] Juan Ramos et al. Using TF-IDF to determine word relevance in document queries. In *Proceedings of the First Instructional Conference on Machine Learning*, volume 242, pages 133–142, 2003.

[239] David Kauchak and CA Claremont. Pomona at Semeval-2016 task 11: predicting word complexity based on corpus frequency. *SemEval*, pages 1047–1051, 2016.

[240] Qinyi Wu, Danesh Irani, Calton Pu, and Lakshmish Ramaswamy. Elusive vandalism detection in Wikipedia: a text stability-based approach. In *Proceedings of the 19th International Conference on Information and Knowledge Management*, pages 1797–1800. Association for Computing Machinery, 2010.

[241] Peter Schönhofen. Identifying document topics using the Wikipedia category network. *Web Intelligence and Agent Systems: An International Journal*, 7:pages 195–207, 2009.

[242] Takeshi Yamada, Kazumi Saito, and Kazuhiro Kazama. Network analyses to understand the structure of Wikipedia. In *Proceedings of Symposium on Network Analysis in Natural Sciences and Engineering*, page 108, 2006.

[243] Sisay Fissaha Adafre and Maarten de Rijke. Discovering missing links in Wikipedia. In *Proceedings of the 3rd international workshop on Link discovery*, pages 90–97. Association for Computing Machinery, 2005.

[244] Sara Javanmardi and Cristina Lopes. Statistical measure of quality in Wikipedia. In *Proceedings of the 1st Workshop on Social Media Analytics*, pages 132–138. Association for Computing Machinery, 2010.

[245] Vasa Hardik, Vasudevan Anirudh, and Palanisamy Balaji. Link analysis of Wikipedia documents using MapReduce. In *Proceedings of International Conference on Information Reuse and Integration 2015*, pages 582–588. Institute of Electrical and Electronics Engineers, 2015.

[246] David Milne. Computing semantic relatedness using Wikipedia link structure. In *Proceedings of the New Zealand Computer Science Research Student Conference*, pages 1–8, 2007.

[247] Tao-Chun Lee and Jayakrishnan Unnikrishnan. Monitoring network structure and content quality of signal processing articles on Wikipedia. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2013.

[248] Andrea Giovanni Nuzzolese, Aldo Gangemi, Valentina Presutti, and Paolo Ciancarini. Encyclopedic knowledge patterns from Wikipedia links. In *Proceedings of the International Semantic Web Conference*, pages 520–536. Springer, 2011.

[249] Bifan Wei, Jun Liu, Qinghua Zheng, Wei Zhang, Chenchen Wang, and Bei Wu.

DF-Miner: Domain-specific facet mining by leveraging the hyperlink structure of Wikipedia. *Knowledge-Based Systems*, 77:pages 80–91, 2015.

[250] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: bringing order to the web. Technical report, Stanford InfoLab, 1999.

[251] Jon Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the Association for Computing Machinery*, 46(5):pages 604–632, 1999.

[252] Dheeraj Rajagopal, Erik Cambria, Daniel Olsher, and Kenneth Kwok. A graph-based approach to commonsense concept extraction and semantic similarity detection. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 565–570. Association for Computing Machinery, 2013.

[253] Zheng Xu, Junyu Xuan, Yunhuai Liu, Kim-Kwang Raymond Choo, Lin Mei, and Chuanping Hu. Building spatial temporal relation graph of concepts pair using web repository. *Information Systems Frontiers*, pages 1–10, 2016.

[254] Zhiyuan Cai, Kaiqi Zhao, Kenny Zhu, and Haixun Wang. Wikification via link co-occurrence. In *Proceedings of the 22nd International Conference on Information & Knowledge Management*, pages 1087–1096. Association for Computing Machinery, 2013.

[255] Stephen Dill, Ravi Kumar, Kevin McCurley, Sridhar Rajagopalan, D Sivakumar, and Andrew Tomkins. Self-similarity in the web. *Association for Computing Machinery, Transactions on Internet Technology*, 2(3):pages 205–223, 2002.

[256] Deepayan Chakrabarti, Yiping Zhan, and Christos Faloutsos. R-MAT: A recursive model for graph mining. In *Proceedings of the 2004 Society for Industrial and Applied Mathematics, International Conference on Data Mining*, pages 442–446. Society for Industrial and Applied Mathematics, 2004.

[257] Sergey Chernov, Tereza Iofciu, Wolfgang Nejdl, and Xuan Zhou. Extracting semantics relationships between Wikipedia categories. *SemWiki*, 206, 2006.

[258] Wikipedia. Wikipedia category taxonomy graph. Provided at http://wikicategory.sourceforge.net/ (last accessed in July 2019), November 2010.

[259] Todd Holloway, Miran Bozicevic, and Katy Börner. Analyzing and visualizing the semantic coverage of Wikipedia and its authors. *Complexity*, 12:pages 30–40, 2007.

[260] Julian Szymański. Mining relations between Wikipedia categories. In *Proceedings of the International Conference on Networked Digital Technologies*, pages 248–255. Springer, 2010.

[261] Jonathan Yu, James Thom, and Audrey Tam. Ontology evaluation using Wikipedia categories for browsing. In *Proceedings of the 16th Conference on Information and Knowledge Management*, pages 223–232. Association for Computing Machinery, 2007.

[262] Cheong-Iao Pang and Robert Biuk-Aghai. Wikipedia world map: method and application of map-like wiki visualization. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration*, pages 124–133. Association for Computing Machinery, 2011.

[263] Qiuju Zhou and Loet Leydesdorff. The normalization of occurrence and co-occurrence matrices in bibliometrics using cosine similarities and ochiai coefficients. *Journal of the Association for Information Science and Technology*, 67(11):pages 2805–2814, 2016.

[264] Jan Buzydlowski, Howard White, and Xia Lin. Term co-occurrence analysis as an interface for digital libraries. In *Collection of Visual Interfaces to Digital Libraries*, pages 133–144. Springer, 2002.

[265] Chi Wang, Marina Danilevsky, Nihit Desai, Yinan Zhang, Phuong Nguyen, Thrivikrama Taula, and Jiawei Han. A phrase mining framework for recursive construction of a topical hierarchy. In *Proceedings of the 19th International Conference on Knowledge Discovery and Data Mining*, pages 437–445. Association for Computing Machinery, 2013.

[266] Harry Halpin, Valentin Robu, and Hana Shepherd. The complex dynamics of collaborative tagging. In *Proceedings of the 16th International Conference on World Wide Web*. Association for Computing Machinery, 2007.

[267] Kevin Boyack, Richard Klavans, and Katy Börner. Mapping the backbone of science. *Scientometrics*, 64:pages 351–374, 2005.

[268] Arzucan Özgür, Burak Cetin, and Haluk Bingol. Co-occurrence network of Reuters news. *International Journal of Modern Physics C*, 19:pages 689–702, 2008.

[269] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the Association for Computing Machinery*, 18(11):pages 613–620, 1975.

[270] Richard Klavans and Kevin Boyack. Identifying a better measure of relatedness for mapping science. *Journal of the American Society for Information Science and Technology*, 57(2):pages 251–263, 2006.

[271] Elizabeth Leicht, Petter Holme, and Mark Newman. Vertex similarity in networks. *Physical Review E*, 73(2), 2006.

[272] Yiye Ruan, David Fuhry, and Srinivasan Parthasarathy. Efficient community detection in large networks using content and links. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1089–1098. Association for Computing Machinery, 2013.

[273] Elnaz Davoodi, Keivan Kianmehr, and Mohsen Afsharchi. A semantic social network-based expert recommender system. *Applied intelligence*, 39(1):pages 1–13, 2013.

[274] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of Wikipedia entities in web text. In *Proceedings of the 15th International Conference, Special Interest Group on Knowledge Discovery and Data Mining*, pages 457–466. Association for Computing Machinery, 2009.

[275] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1375–1384. Association for Computational Linguistics, 2011.

[276] Mohamed Ben Aouicha, Mohamed Ali Hadj Taieb, and Malek Ezzeddine. Derivation of 'is a' taxonomy from Wikipedia category graph. *Engineering Applications of Artificial Intelligence*, 50:pages 265–286, 2016.

[277] David S Johnson, Cecilia R Aragon, Lyle A McGeoch, and Catherine Schevon. Optimization by simulated annealing: an experimental evaluation; part i, graph partitioning. *Operations research*, 37(6):pages 865–892, 1989.

[278] George Karypis and Vipin Kumar. A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. *University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center, Minneapolis, MN*, 1998.

[279] Inderjit S Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh International conference on Knowledge discovery and data mining*, pages 269–274. Association for Computing Machinery, 2001.

[280] Thorsten Joachims. Transductive learning via spectral graph partitioning. In *Proceedings of the 20th International Conference on Machine Learning*, pages 290–297, 2003.

[281] Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. *Communications of the Association for Computing Machinery*, 51(1):pages 107–113, 2008.

[282] U Kang, Hanghang Tong, Jimeng Sun, Ching-Yung Lin, and Christos Faloutsos. Gbase: a scalable and general graph management system. In *Proceedings of the 17th International Conference Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining*, pages 1091–1099. Association for Computing Machinery, 2011.

[283] Grzegorz Malewicz, Matthew Austern, Aart Bik, James Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. Pregel: a system for large-scale graph processing. In *Proceedings of the 2010 International Conference Association for Computing Machinery's Special Interest Group on Management of Data*, pages 135–146, 2010.

[284] Lu Wang, Yanghua Xiao, Bin Shao, and Haixun Wang. How to partition a billion-node graph. In *Proceedings of the 30th International Conference on Data Engineering 2014*, pages 568–579. Institute of Electrical and Electronics Engineers, 2014.

[285] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *Society for Industrial and Applied Mathematics Journal on scientific Computing*, 20(1):pages 359–392, 1998.

[286] Bruce Hendrickson and Tamara G Kolda. Graph partitioning models for parallel computing. *Parallel computing*, 26(12):pages 1519–1534, 2000.

[287] TS Evans and Renaud Lambiotte. Line graphs, link partitions, and overlapping communities. *Physical Review E*, 80(1):page 016105, 2009.

[288] Hongyuan Zha, Xiaofeng He, Chris Ding, Horst Simon, and Ming Gu. Bipartite graph partitioning and data clustering. In *Proceedings of the 10th international conference on Information and knowledge management*, pages 25–32. Association for Computing Machinery, 2001.

[289] Wook-Shin Han, Sangyeon Lee, Kyungyeol Park, Jeong-Hoon Lee, Min-Soo Kim, Jinha Kim, and Hwanjo Yu. Turbograph: a fast parallel graph engine handling billion-scale graphs in a single pc. In *Proceedings of the 19th International Conference Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data mining*, pages 77–85. Association for Computing Machinery, 2013.

[290] Aapo Kyrola, Guy Blelloch, and Carlos Guestrin. Graphchi: Large-scale graph computation on just a pc. USENIX, 2012.

[291] George Karypis and Vipin Kumar. Multilevel algorithms for multi-constraint graph partitioning. In *SC'98: Proceedings of the 1998 Association for Computing Machinery/Institute of Electrical and Electronics Engineers, Conference on Supercomputing*, pages 28–28. Institute of Electrical and Electronics Engineers, 1998.

[292] George Karypis and Vipin Kumar. Multilevelk-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed computing*, 48(1):pages 96–129, 1998.

[293] Xiaoli Zhang Fern and Carla E Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the twenty-first international conference on Machine learning*, page 36. Association for Computing Machinery, 2004.

[294] Isabelle Stanton and Gabriel Kliot. Streaming graph partitioning for large distributed graphs. In *Proceedings of the 18th International Conference on Knowledge discovery and data mining*, pages 1222–1230. Association for Computing Machinery, 2012.

[295] Chris Walshaw, Mark Cross, and Martin G Everett. Parallel dynamic graph partitioning for adaptive unstructured meshes. *Journal of Parallel and Distributed Computing*, 47(2):pages 102–108, 1997.

[296] Yaroslav Akhremtsev, Peter Sanders, and Christian Schulz. High-quality shared-memory graph partitioning. In *European Conference on Parallel Processing*, pages 659–671. Springer, 2018.

[297] Toby Simpson, Dimosthenis Pasadakis, Drosos Kourounis, Kohei Fujita, Takuma Yamaguchi, Tsuyoshi Ichimura, and Olaf Schenk. Balanced graph partition refinement using the graph p-laplacian. In *Proceedings of the Platform for Advanced Scientific Computing Conference*, page 8. Association for Computing Machinery, 2018.

[298] Jim Webber. A programmatic introduction to neo4j. In *Proceedings of the 3rd Annual Conference on Systems, Programming, and Applications: Software for Humanity*, pages 217–218. Association for Computing Machinery, 2012.

[299] Vladimir Batagelj, Andrej Mrvar, and Matjaž Zaveršnik. Partitioning approach to visualization of large graphs. In *Proceedings of the International Symposium on Graph Drawing*, pages 90–97. Springer, 1999.

[300] George Davidson, Bruce Hendrickson, David Johnson, Charles Meyers, and Brian Wylie. Knowledge mining with Vxinsight: Discovery through interaction. *Journal of Intelligent Information Systems*, 11(3):pages 259–285, 1998.

[301] Kazumi Saito, Takeshi Yamada, and Kazuhiro Kazama. Extracting communities from complex networks by the k-dense method. *The Institute of Electronics,*

Information and Communication Engineers, Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 91(11):pages 3304–3311, 2008.

[302] Bruce Hendrickson and Tamara Kolda. Graph partitioning models for parallel computing. *Parallel computing*, 26(12):pages 1519–1534, 2000.

[303] Harry Halpin. *Social semantics: the search for meaning on the web*. Springer Publishing Company, Incorporated, 2012.

[304] Heiko Paulheim and Christian Bizer. Type inference on noisy RDF data. In *Proceedings of International Semantic Web Conference*, pages 510–525. Springer, 2013.

[305] Sergey Chernov, Tereza Iofciu, Wolfgang Nejdl, and Xuan Zhou. Extracting semantics relationships between Wikipedia categories. *SemWiki*, 206, 2006.

Appendix A

Giant Clusters Split

This appendix contains further visualisations of the clusters presented in Chapter 5, Section 5.4, plotted using Pajek. The figures are the representation of the category clusters results for the English editions 2010 to 2012 and 2015. Each figure shows the clusters in global (the first row) and local (the second row) views where the giant cluster split into two clusters for the critical weight thresholds between t and t+1.

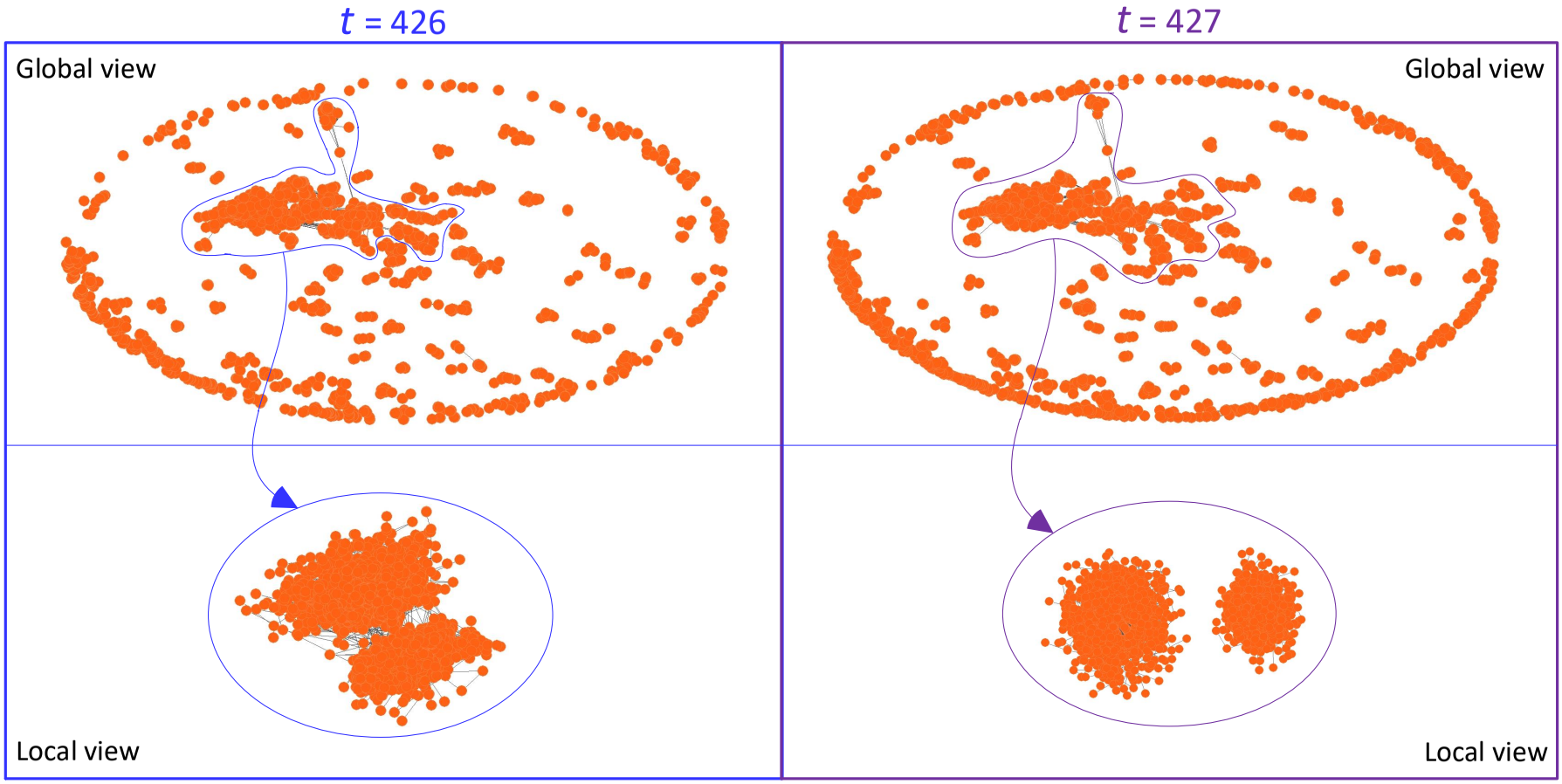


Figure A.1 Visualisation-global and local views of giant category clusters split between weight threshold 426 and 427 for English Wikipedia category co-occurrence network 2010

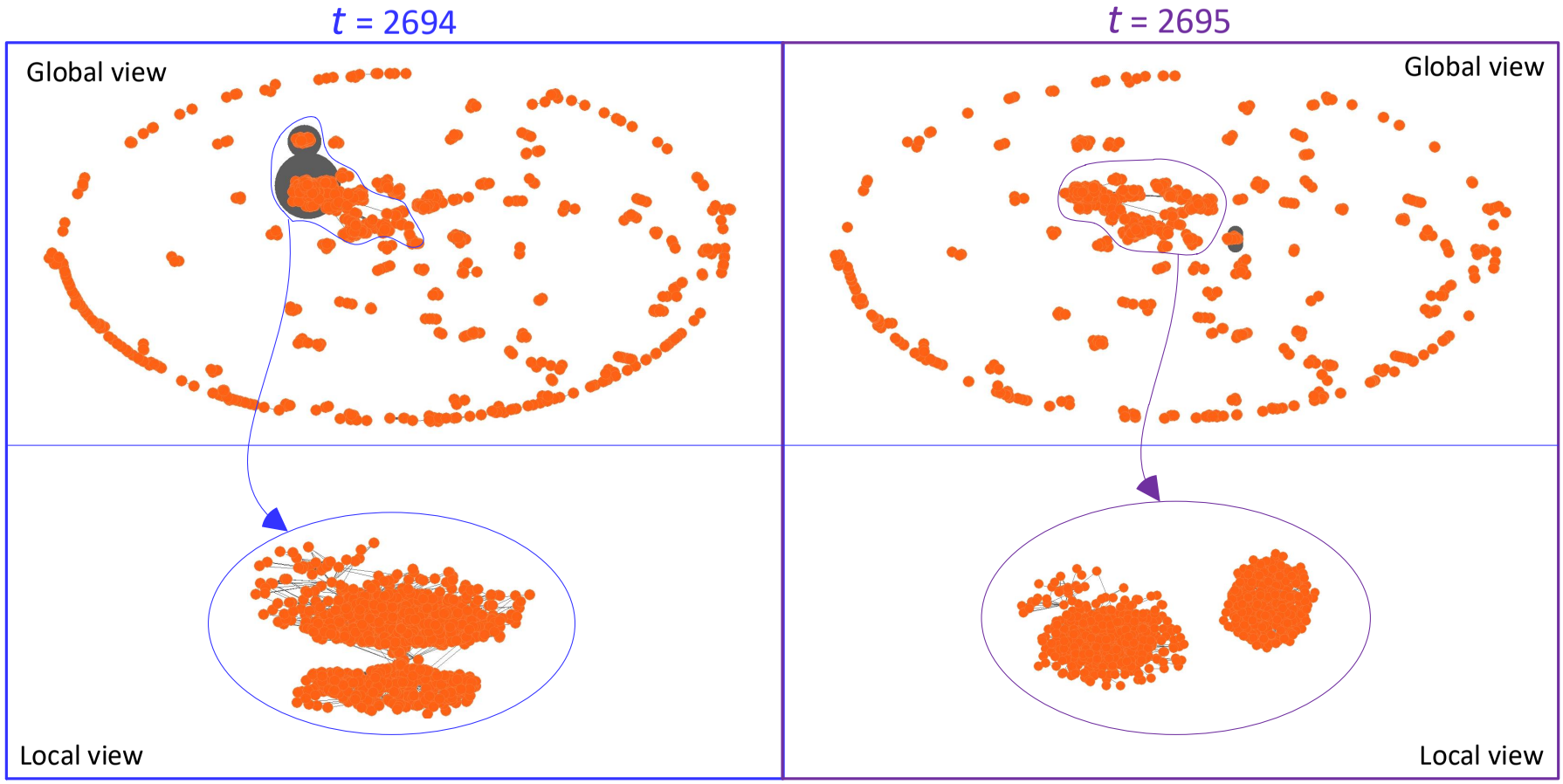


Figure A.2 Visualisation-global and local views of giant category clusters split between weight threshold 2694 and 2695 for English Wikipedia category co-occurrence network 2011

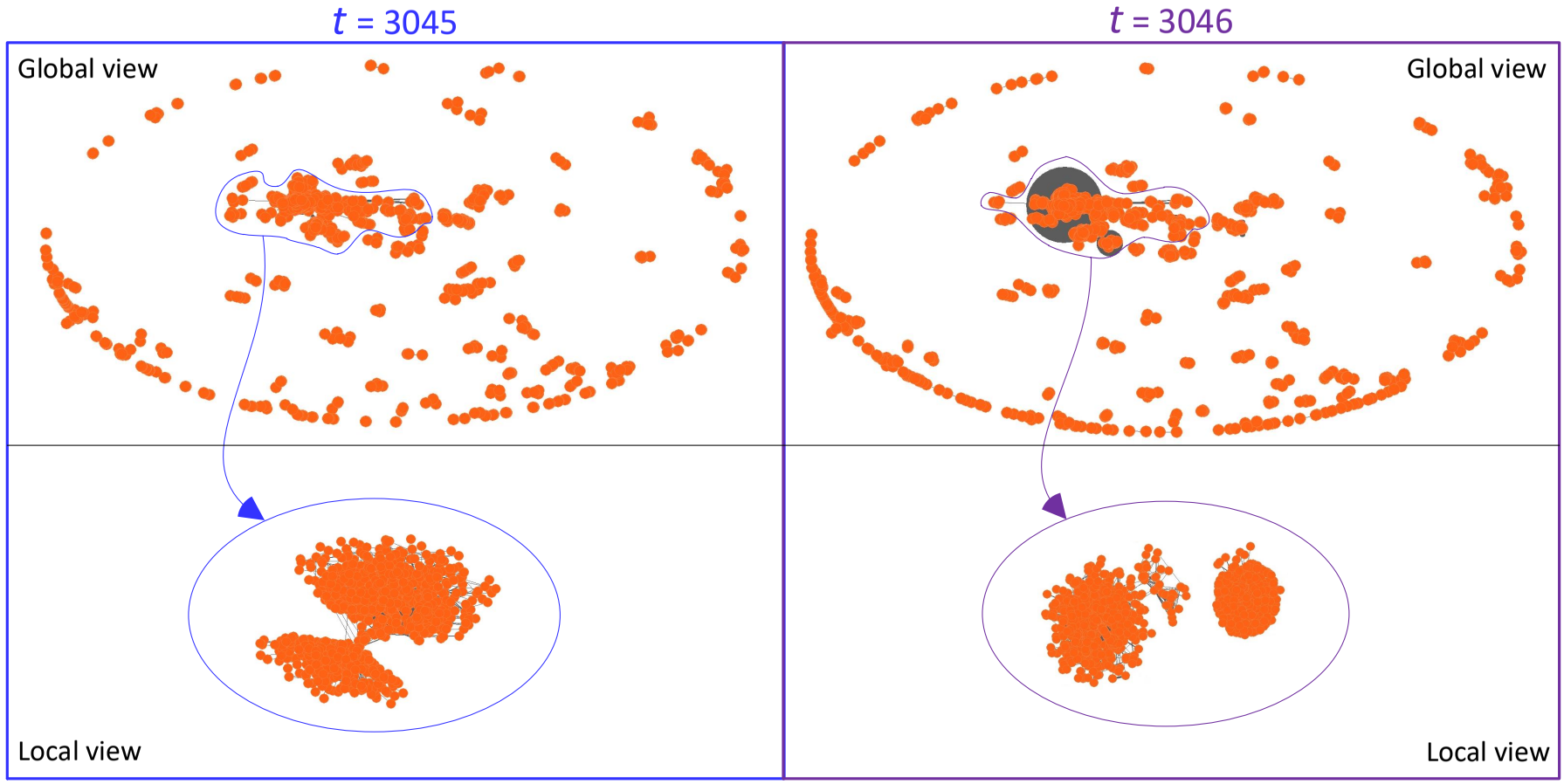


Figure A.3 Visualisation-global and local views of giant category clusters split between weight threshold 3045 and 3046 for English Wikipedia category co-occurrence network 2012

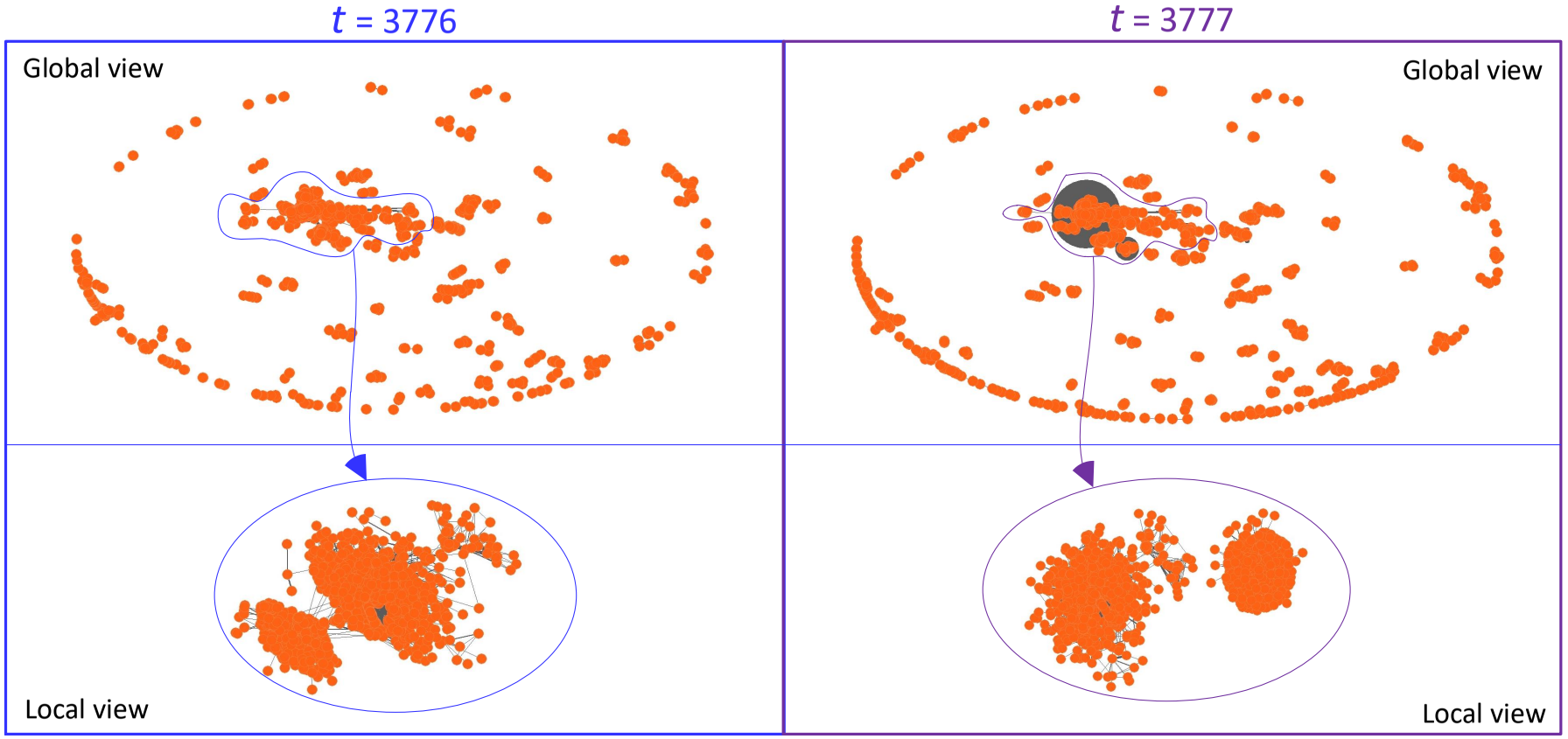


Figure A.4 Visualisation-global and local views of giant category clusters split between weight threshold 3776 and 3777 for English Wikipedia category co-occurrence network 2015

Appendix B

German Editions

This appendix presents the further results on the German Wikipedia networks performed using the *t*-component framework found in Chapter 5. The figures in the next pages show the main four observed properties in the clusters size for the Wikipedia German editions 2010 to 2012. Each figure shows the finding for each graph property against varying weight thresholds.

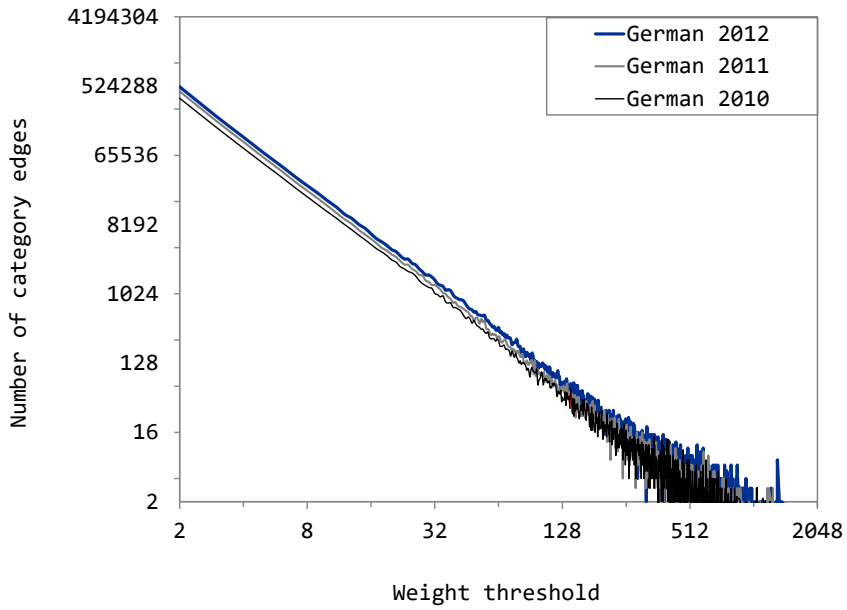


Figure B.1 Log-log plots-the number of category edges for different weight threshold values for German category co-occurrence networks 2010-2012

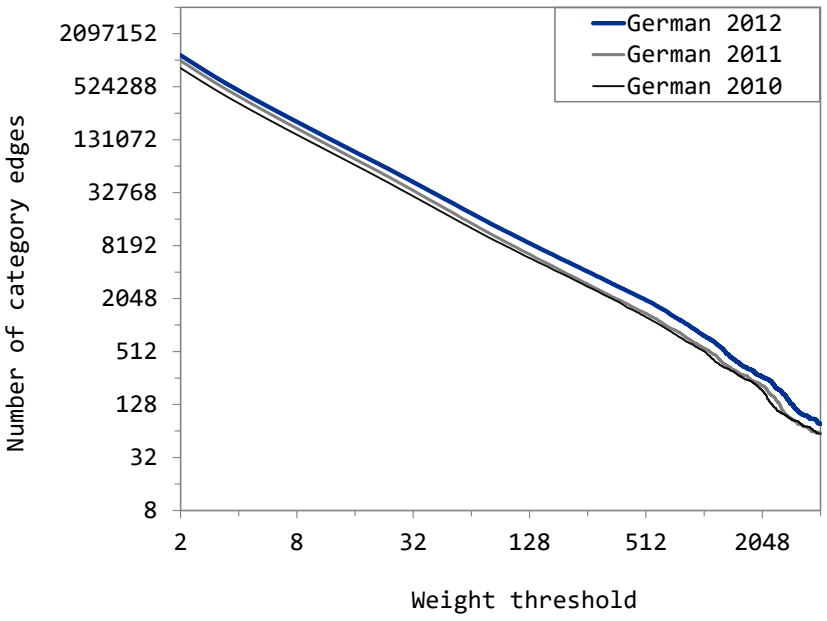


Figure B.2 Log-log plots-the cumulative number of category edges for different weight threshold values for German category co-occurrence networks 2010-2012

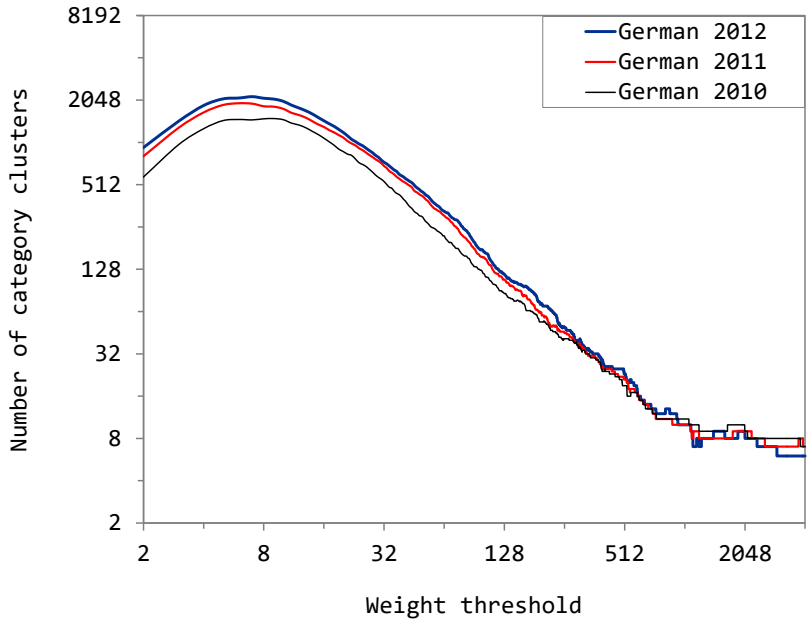


Figure B.3 Log-log plots-the number of category clusters for different weight threshold values for German category co-occurrence networks 2010-2012

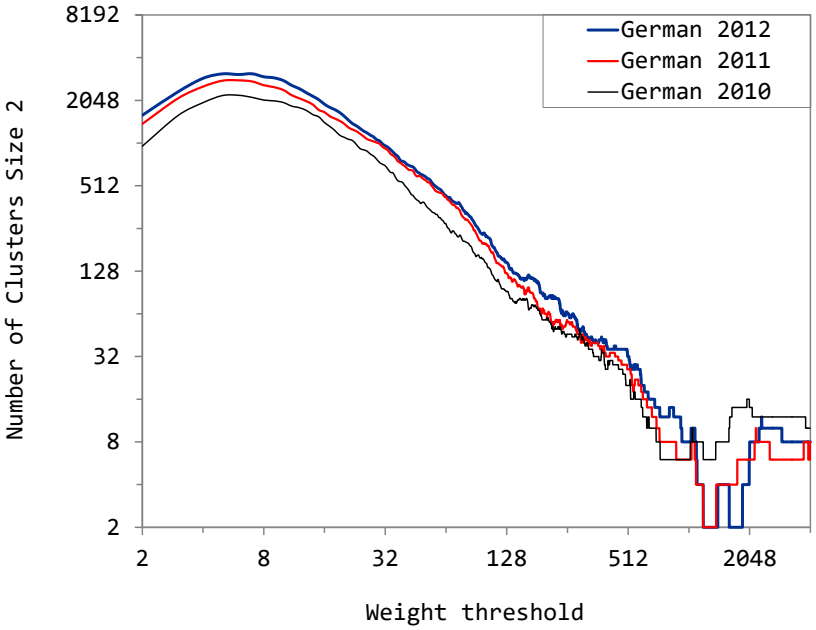


Figure B.4 Log-log plots-the number of cluster size two for different weight threshold values for German category co-occurrence networks 2010-2012

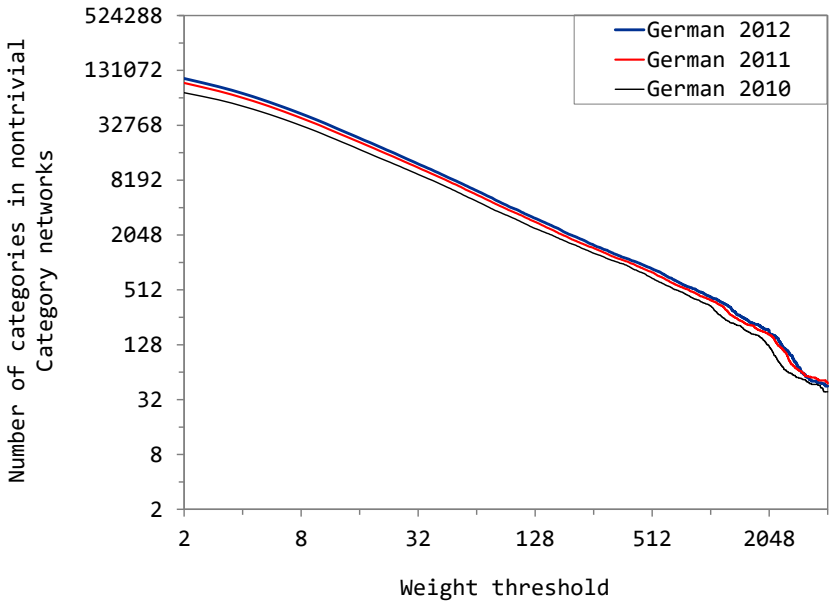


Figure B.5 Log-log plots-the number of categories for different weight thresholds for the German Wikipedia category networks from 2010 to 2012

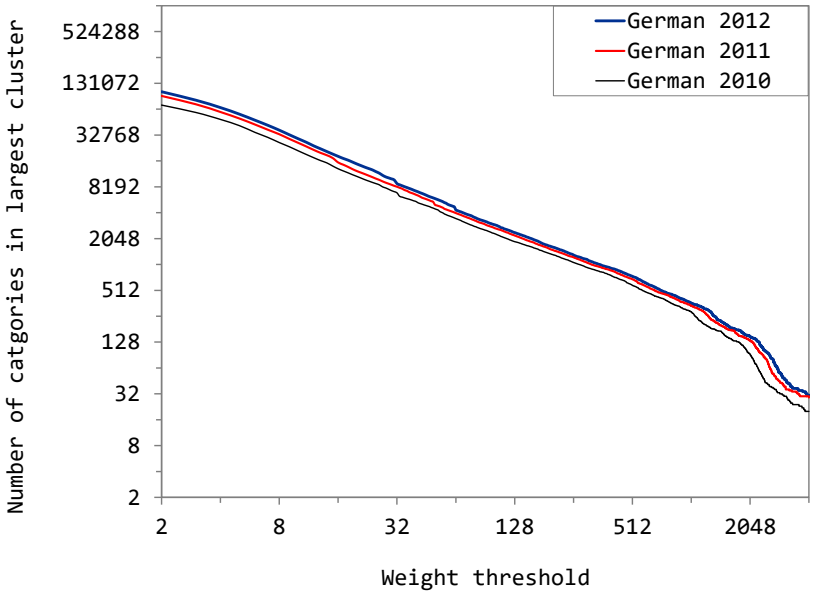


Figure B.6 Log-log plots-the size of the largest cluster for different weight threshold values for German category co-occurrence networks 2010-2012